International Journal of Computational and Experimental Science and ENgineering (IJCESEN)
Vol. 7-No.2 (2021) pp. 84-94
http://dergipark.org.tr/en/pub/ijcesen

ISSN: 2149-9144

Research Article

Comprehensive Analysis of Forest Fire Detection using Deep Learning Models and Conventional Machine Learning Algorithms

Süha Berk KUKUK[1], Zeynep Hilal KİLİMCİ[2, *]

[1]Gedik University, Engineering Faculty, Department of Mechatronic Engineering, 34876, İstanbul-Turkey
shbkukuk@gmail.com-- **ORCID:** 0000-0003-1651-2417--

[2]Kocaeli University, Technology Faculty, Department of Information Systems Engineering, 41001, Kocaeli-Turkey
* **Corresponding Author**: zeynep.kilimci@kocaeli.edu.tr -- **ORCID:** 0000-0003-1497-305X

Article Info:

DOI: 10.22399/ijcesen.950045
Received : 09 June 2021
Accepted : 06 July 2021

Keywords

Forest Fire Detection
Deep learning
Machine Learning
Object Detection
Convolutional Neural Networks

Abstract:

Forest fire detection is a very challenging problem in the field of object detection. Fire detection-based image analysis have advantages such as usage on wide open areas, the possibility for operator to visually confirm presence, intensity and the size of the hazards, lower cost for installation and further exploitation. To overcome the problem of fire detection in outdoors, deep learning and conventional machine learning based computer vision techniques are employed to determine the fire detection when indoor fire detection systems are not capable. In this work, we propose a comprehensive analysis of forest fire detection using conventional machine learning algorithms, object detection techniques, deep and hybrid deep learning models. The contribution of this work to the literature is to analyze different classification and object detection techniques in more details that is not addressed before in order to detect forest fire. Experiment results demonstrate that convolutional neural networks outperform other methods with 99.32% of accuracy result.

1. Introduction

Forest fire is unplanned, uncontrolled, unwanted fire what start natural or people cause. Forest fire results in a huge number of global disasters, causing endless economic and ecological losses as well as endangering living population. Every year, after all, millions of hectares of forest are destroyed by forest fire, which seriously just lost source oxygen and rise carbon dioxide cause change earth climate. In order to guard living population and preserve natural resource fire detection and fighting with fire have been seen as one of the most important solution. Traditional fire detect systems are heat detectors, smoke detectors, carbon monoxide detectors and manual call points. All of fire detect systems don't successfully work outdoors because they because are designed to work indoors.

Forest fire detection is a very challenging problem computer vision in the field of object detection. Fire detection-based image analysis have advantages such as usage on wide open areas, the possibility for operator to visually confirm presence, intensity and the size of the hazards, lower cost for installation and further exploitation. To overcome the problem of fire detection in outdoors, deep learning and conventional machine learning based computer vision techniques are employed to determine the fire detection when indoor fire detection systems are not capable. [1-3].

In addition to traditional machine learning methods, the application of deep learning algorithms has become very popular in recent years in different research areas such as image processing [2], natural language processing [4], speech recognition [5], and machine translation [6], computer vision [7], etc. Deep learning models are opted by scientists and researchers since they present better forecasts when compared to conventional machine learning techniques. Deep learning models are employed for the purpose of ensuring automatic feature extraction by training complex features to acquire more informative demonstration of data. In addition to this, deep learning methods are also utilized employed for the classification tasks in many areas

[8-10]. Convolutional neural networks (CNNs), recurrent neural networks (RNNs), long short-term memory networks (LSTMs), generative adversarial Networks (GANs), radial basis function networks (RBFs), and deep belief networks (DBNs) are widely used and well-known architectures. Both traditional machine learning techniques and deep learning methodologies are employed to detect fire indoors and outdoors [11-15].

In this paper, we propose an efficient forest fire detection model by using traditional machine learning algorithms, deep learning models, hybrid deep learning methodologies and object detection techniques (OD) in order to demonstrate the most successful system by comparing the performance of them. For this purpose, support vector machines (SVMs) and random forest (RF) models are evaluated as conventional machine learning techniques. CNN as deep learning model, convolutional neural network-gated recurrent unit (CNN-GRU) and convolutional neural network-long short-term memory (CNN-LSTM) as hybrid deep learning techniques, and Single Shot Detector (SSD), faster region based convolutional neural network (Faster R-CNN) as OD methodology are assessed. In order to demonstrate the efficiency of proposed model, two different the data sets are collected from two different sources. DL techniques have changeable hyper parameters that are the better get result for models. We also attempt model's different version used that disparate hyperparameters to acquire the best performance by evaluating eight different combination every model of hyper parameters. The results of extensive experiment indicate that CNN model exhibits superior classification performance in order to detect forest fire compared to all other techniques. To the best of our knowledge, this is the very first attempt in terms of utilizing and comparing the performance of various models such as traditional machine learning algorithms, deep learning models, hybrid deep learning methodologies, object detection techniques.

The rest of this article is arranged as follows: Section 2 provides a summary of studies on fire detection. Section 3 includes methods used in this work, including machine learning techniques deep learning, hybrid deep learning and object detection models. Section 4 introduces proposed framework. The experiment results, discussion and conclusion parts are given in Sections 5 and 6, respectively.

2. Related Work

This section provides a brief summary of the state-of-the-art studies on fire detection. In [7], Rafiee et al. investigate which object was smoke and fire by taking all objects in an image. Authors perform detection of smoke using color, mobility and tissue disorder characteristics. In [8], Ren et al. present performance of support vector machine (SVM) algorithm as a machine learning algorithm for fire classification in open space using color channels. They report that it is possible to calculate which color channel is outweighed in the image and whether it contains fire in the image according to the four main rules set in the color channels special. Data sets consisting of video images are labelled. F-measure is employed as an evaluation metric to demonstrate the success of support vector machine technique. They conclude the paper that the usage of SVM ensures 93.52% classification success.

In [9], Sadewa et al. propose to reduce the time elapsed from the start of the fire to the shortest time so that smoke sensors can test smoke indoors. For this purpose, the data received by the web cam placed at a corner point of the room instead of the smoke sensor is gathered. The data is first preprocessed by image processing techniques and then model training is carried out using evolutionary neural network. Inception-v3 model that is a convolutional neural network architecture from the Inception family and bottleneck features to classify class of fire. In [10], a real time fire detector is developed using faster region-based convolutional neural networks (Faster R-CNN). For fire detection system process of the machine learning, 1,000 images which are comprise fire and forest scenes are splitted into training (80%) and validation sets (20%). Experiment results demonstrate that Faster R-CNN is capable to detect fire and forest scene with 99% of accuracy. In [11], Mwedzi et al. introduce an intelligent fire detection system procedure with the use of computer vision and convolutional neural network to analyze the performance of the fire detection system under various setup scenarios. CNN and VGG-net models are used for fire detection system. The data set is constructed by using Google image search. The efficiency of the trained neural network data set of images are classified based on the precise images at the test stage. The experiment results are demonstrated that the proposed intelligent fire detection system presents significant improvement as compared to non-intelligent mechanisms. Through the application of CNN and VGGNet models, the system performs better result with 85% of accuracy and minimizes false alarm. In [13], Xu et al, propose to detect forest fire by employing deep learning methodology. Yolov5 and EfficientDet are utilized for fire detection process. The paper was created a single integrated forest fire dataset containing 10,581 images, with 2976 forest fire images and 7605 non-fire images. Experiments on the dataset show that the proposed method improves detection performance by 2.5% to 10.9%, and decreases false positives by 51.3%, without any extra latency. In [14], Pourghasemi et al. assess forest fire detection sensitivities in Fars Province, Iran. Three geographic information system-based machine learning techniques are utilized, namely regression tree, general linear model and compound discriminant test. Data set is composed of satellite images on forest. 358 sites in the data set are divided into two sets for training (70%) and validation (30%). The BRT model is got access of accuracy is 83%. They conclude the paper that resulting maps can enhance the effectiveness of planning and management of forest resources and ecological balances in this province area. In [15], Maksymiv et al. focus on fire detection by combining some advantages of the AdaBoost, local binary pattern (LBP) and convolutional neural network for the purpose of both accelerating the processing time and providing better performance. Execution time is in the millisecond range, thus verifying that the developed system can operate in real time at video rates. In addition, the utilization of CNN technique exhibits more accurate classification success. The performance of CNN is validated on the data set with five hundred images, which contain such category as smoke, fire, others. The empirical results show that the method achieves more than 95% correct detection rate.

Our study differs in literature studies aforementioned in that it enables to evaluate various machine learning algorithms, deep and hybrid learning algorithms, and object detection model. To our knowledge, it is the first study to evaluate forest fire in such a comprehensive way in terms of six different techniques and two different size data sets.

3. Models

In this section, methods used in this work, including machine learning techniques, deep and hybrid deep learning and object detection models are briefly presented.

3.1. Support Vector Machine (SVM)

The support vector machine (SVM) classifier, which is supervised learning, one of the most accurate and robust algorithms in data collection and analyze data for classification and regression analysis. The SVM algorithm was developed by Cortes and Vapnik in 1995 [16] to classify data that were linearly separable and later generalized to nonlinear states. When using SVM training, data are categorized into two phases of train and test sets. and to validate the data, cross-validation techniques such as k-fold, holdout, or leave-n-out training methods are applied. In addition to performing linear classification, SVMs can efficiently occur a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces [17].

3.2. Random Forest (RF)

Random forest (RF) or random decision forests are an ensemble learning technique for classification and regression. The general machine learning method of random decision forests was first proposed by Ho in 1995 [18]. Random Forest is a supervised learning algorithm. It is used in classification and regression problems with random decision forests. The forest network structure he established is a collection of decision trees, often trained by the bagging method. The Bagging method is that a combination of learning models increases the overall result. The random forest forms multiple land trees and combines these trees for a more accurate and stable prediction. Random Forest adds additional randomness to the model as it grows trees. Instead of looking for the most important property when breaking a node apart, it looks for the best property among a random subset of properties. This results in a wide variety, which usually results in a better model. Thus, in a random forest, only a random subset of properties is considered by the algorithm for the division of a node. Instead of finding the best possible thresholds, the model creates decision trees using random thresholds for each individual property.

3.3. Convolution Neural Network. (CNN)

CNNs are a particular kind of deep learning algorithm [19] for most commonly applied to analyze visual imagery that give a better consequence than many other machine learning algorithms. CNNs are regularized versions of multilayer perceptron. A CNN include input layer, hidden layer and output layer. Hidden layer is provided that the better learning capability of CNN and can learn from the data. CNN is also a feedforward Neural Network with more hidden layers. Hidden layers comprise convolutional layers combined with pooling layers. CNN has the most important block that is convolutional layer. The input is a tensor. After passing through a convolutional layer, the data becomes abstracted to a feature map. Multilayer filters are applied for the output is given to batch of feature map that becomes the final output of the convolutional layer.

Convolution layer catch information about local dependencies or semantics in the regions of original data. Data and final output are masked by the activation function. After the convolution process, a pooling layer transform the number of samples in each feature map and take the most important information. Pooling layers transform the size of data by compound the outputs of neuron bunch at one layer into a single neuron in the next layer. There are two common types of pooling in popular use: max and average. Max pooling uses the maximum value of each local cluster of neurons in the feature map, while average pooling takes the average value.

3.4 Convolution Neural Network-Gated Recurrent Unit (CNN-GRU)

Gated recurrent unit (GRU) is improved for that are solved the vanishing gradient problem which come from with standard recurrent neural networks (RNNs) by Cho et al. [19]. GRU is a variation of long short-term memory networks (LSTMs). The GRU is simpler than the LSTM since it has only two gates, which are reset gate and update gate. These gates are utilized for the purpose of evaluating which information is beneficial or not. GRU has two gates that are update and reset aforementioned before. The useful information is stored with the usage of update gate while useless information is forgotten by reset gate.

The hybrid CNN-GRU model contains the best advantages of the GRU model which can be processed time sequence data and the advantages of the CNN model which is ideal for handling high dimensional data and feature extraction process. The architecture of the proposed GRU-CNN hybrid neural networks consists of structured GRU module and a CNN module. The inputs are fire and forest images data collected from web scraping while the outputs are the prediction of the is fire or non-fire. The CNN module uses local connection and shared weights to directly extract local features from the images matrixes data and obtain effective representation through the convolution layer and pooling layer. On the other hand, the purpose of the GRU module is to prevent the long-term dependency. In this way, the GRU module learns useful information in the historical data for a long period through the memory cell, and the useless information is forgotten by the forget gate. The inputs of GRU module are time sequence data; the GRU module includes many gated recurrent units, and the outputs of all these gated recurrent units are connected with the fully connected layer. Finally, the load predicting results can be obtained by calculating the mean value of all neurons in the fully connected layers.

3.5. Convolutional Neural Network-Long Shor-Term Memory (CNN-LSTM)

Long short-term memory networks (LSTMs) is an extended version of the recurrent neural networks (RNNs) architecture [17]. Unlike forward-fed neural networks, LSTM has feedback connections. These connections can flow information to the next neuron with the help of gates, or this information can be considered unnecessary and forgotten. An ordinary LSTM unit consists of a cell, an entrance gate, an exit gate and a forget gate. The cell remembers values at arbitrary intervals of time, and these three gates regulate the flow of information entering and exiting the cell. The performance of LSTMs is very poor compared to other deep learning algorithms such as CNNs in computer vision or image classification problems [20]. For this reason, hybrid models of LSTMs are evaluated to enhance the classification success in these fields [21]. Due to the characteristics of CNN and LSTM, the combination of advantages of CNN and LSTM is presented as a hybrid model. In this study, we also employ a hybrid deep learning methodology by integrating CNN and LSTM techniques. Two layers of CNN are utilized to ensure the correlation and effective extraction of multidimensional data. The feature sequence from the CNN layer is considered as the input for LSTM. Finally, the time dependencies are also extracted in the LSTM layer.

3.6. Faster Recurrent-Convolutional Neural Network (Faster R-CNN)

Faster R-CNN is model of R-CNN technology. The model is improved by Girshick et al [22]. A Faster R-CNN object detection network is composed of a feature extraction network which is typically a pretrained CNN. This is then followed by two subnetworks which are trainable. The first is a Region Proposal Network (RPN), which is, as its name suggests, used to generate object proposals and the second is used to predict the actual class of the object. So, the primary differentiator for Faster R-CNN is the RPN which is inserted after the last convolutional layer. This is trained to produce region proposals directly without the need for any external mechanism like selective search. Then, region of interest (ROI) pooling and an upstream classifier and bounding box regressor similar to Fast R-CNN are employed for the purpose of detecting fire at outdoor. We also use ROI pooling and an upstream classifier and bounding box regressor similar to Fast R-CNN.

3.6. Single Shot Detector (SSD)

The SSD model is a model used in object detection, such as the Faster R-CNN model. In contrast to the R-CNN architecture, the SSD model performs these operations at the same time, while the region specified in the networks and these regions are connected to the networks respectively. This feature allows you to achieve faster results than the Faster R-CNN model of the SSD model. The input image is to the SSD model as in other models. It passes through convolutional nerve layers. The dimensions of these layers are different. In this way, different feature maps of the image are extracted. A certain number of rectangles are obtained using a 3x3 convolutional filter on maps. SSD divides the image as grids, and each grid cell responsible for detecting objects in that region of the image [23].

4. Proposed Framework

In this work, the comprehensive analysis of forest fire detection is proposed by employing deep and hybrid deep learning models, conventional machine learning algorithms, and object detection algorithms. For this purpose, data collection procedure is performed as a first step. The fire image data used in this study is gathered from Google images, Shutterstock, Getty Images websites by data scraping method. The pages of these websites are created with text-based markup languages. The markup language helps to define the structure of arguments that a web page contains. Because there are universal components and tags for embedded data on the website, it allows us to get the information that web breakers need into the program. In this work, a bot is constructed for the purpose of finding the image data addresses inside the website, and taking a screenshot of each image using Selenium library written in Python programming language. In this way, by opening a web browser with the help of a driver that will be installed on the computer, all operations that can be performed in a web browser are performed with the help of a program that conducts Python codes. Another method of scraping data is the usage of Beautiful Soup library, is a powerful and fast library built for processing HTML or XML files. On the web page that hosts the image data, the keyword that will allow to find the image cluster data to search is entered into the program. By parsing the HTML codes in the source with this module, it downloads only the source addresses of the target specified fields. As a result of scraping data, two datasets are gathered. The first one (DS1) covers 2,360 forest fire and 2,400 forest images. Second one (DS2) contains two categories 5,000 forest fire and 5,000 only forest images. After data collection, images are preprocessed by using grayscale and resize methods for both datasets. After that, dataset is ready to model for the purpose of detecting forest fire. For this purpose, support vector machines (SVMs) and random forest (RF) are evaluated as conventional machine learning algorithms while Convolutional Neural Networks (CNNs), Convolutional Gated Recurrent Unit (CNN-GRU), and Convolutional Neural Network-Long Shor-Term Memory (CNN-LSTM) are assessed as deep end hybrid deep learning techniques. In addition, the effect of object detection techniques is also observed by employing Faster Recurrent-Convolutional Neural Network (Faster R-CNN) and Single Shot Detector (SSD) methods. As a result of extensive experiments, the final decision of the forest fire detection is determined among all techniques. The flowchart of the proposed framework is presented in Figure 1. As seen in in Figure 1, data collection step is carried out by gathering images from Google images, Shutterstock, Getty Images websites as a first step. Then, images are classified as forest, and forest fire. After applying grayscale and resizing preprocessing stages, classification task is performed using SVM, RF, CNN, CNN-GRU, CNN-LSTM techniques. In addition to classification models, SSD and Faster R-CNN are employed as object detection models. As a final decision, CNN outperforms others among classification techniques while SSD performs well when compared to the Faster R-CNN.

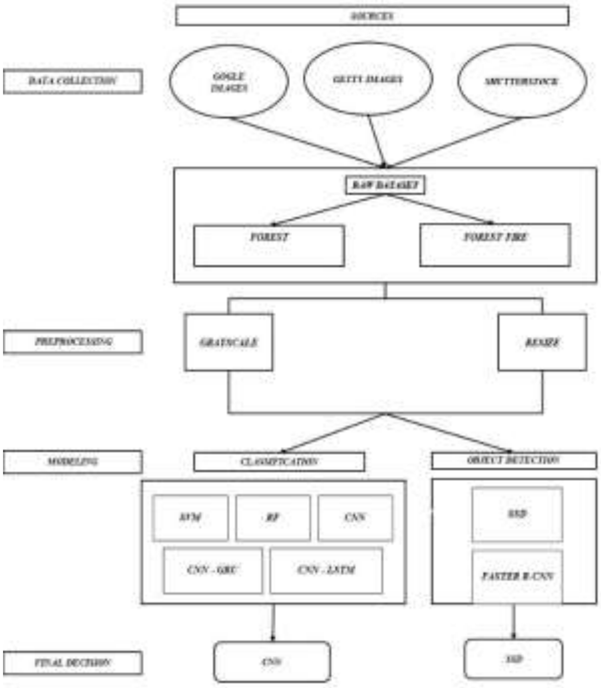

***Figure 1**. The flowchart of the proposed framework*

In this part, parameter details of models are explained. Support vector machines (SVMs) that are machine algorithm, are used kernel function parameters, that represents the similarity of vectors (training samples) in a feature space over polynomials of the original variables, allowing learning of non-linear models [24]. Our model is non-linear models that is so we are used polynomial kernel in SVM model. Other parameter is probability. One standard way to obtain a “probability” out of an SVM is to use Platt scaling, which is available in many decent SVM implementations. In the binary case, the probabilities are calibrated using Platt scaling: logistic regression on the SVM’s scores, fit by an additional cross-validation on the training data. Random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. The sub-sample size is controlled with the max_samples parameter, otherwise the whole dataset is used to build each tree. Our model is used default value. Convolution neural network (CNN) model is deep learning algorithm that contains many parameters of learning models. The convolution layer is the core building block of a convolutional network that does most of the computational heavy lifting. The convolution layer’s parameters consist of a set of learnable filters. Every filter is small spatially (along width and height), but extends through the full depth of the input volume. During the forward pass, we slide (more precisely, convolve) each filter across the width and height of the input volume and compute dot products between the entries of the filter and the input at any position. As we slide the filter over the width and height of the input volume, we will produce a 2-dimensional activation map that gives the responses of that filter at every spatial position.

Intuitively, the network will learn filters that activate when they see some type of visual feature such as an edge of some orientation or a blotch of some color on the first layer, or eventually entire honeycomb or wheel-like patterns on higher layers of the network. Convolutional networks may include local and global pooling layers along with traditional convolutional layers. Pooling layers decrease the size of data by combining the outputs of neuron bunch at one layer into a single neuron in the next layer. Local pooling combines small bunches, tiling sizes such as 2 x 2 which is employed in this work, are commonly used. Global pooling actions on all the neurons of the feature map. There are two common types of pooling in popular use: max and average. Max pooling uses the maximum value of each local cluster of neurons in the feature map, while average pooling takes the average value. Our model is used max pooling layer. Flatten layers for are used, neural networks retrieve input data from a one-dimensional array. The data in this neural network is a one-dimensional array of matrices from the convolutional and pooling layer [20].

Until this part, parameters of convolutional neural networks are mentioned in hybrid models (CNN-LSTM, CNN-GRU), all the parameters mentioned at the top are used in common. From here on, the parameters used in all general models are mentioned. The batch size is a number of samples processed before the model is updated. Batch size are chosen 32 for CNN, 8 for hybrid deep learning algorithms and 4 for object detection models. The number of epochs is the number of complete passes through the training dataset. Activation functions are used to get the output of node. Epoch value is changeable for every models. Epoch size is set to 20 for CNN, 50 for hybrid deep models, and 20,000 for object detection models. The resulting values range between 0 to 1 or -1 to 1. We employed ReLU activation function. We use loss function is Binary cross entropy. Adam is used optimizer for method of machine learning. Parameters that are chosen above were first studied with two different data set sizes and their results were compared. In order to see which hyper parameter variable the model receives the best result, different combinations of the parameters convolution layer number, the number of neurons of the hidden layer and the number of dense layers were tried and the results of the model training criteria were obtained. The parameter values are selected because the best classification results are obtained. The result is argued section 5.

Python 3.7 version is employed in the experiments. Furthermore, pillow, lxml, Tensorboard, OpenCV, TensorFlow, NumPy and Scikit-Learn libraries are integrated into the system for use in the workspace through Anaconda. TensorFlow contains objects detection API (Faster R-CNN, SSD), machine learning tools (Dense, Layer, Sequential) that's so our models are used TensorFlow for machine learning applications. It is a library for the Python programming language, adding support for large, multidimensional arrays and matrices, as well as a large collection of high-level mathematical functions to work with in these arrays. It is used to translate images into tensors. Scikit-learn is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support vector machines, random forests and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy. Scikit-learn is used for our models that SVM and random forest classifier. Python Imaging Library is a free and open-source additional library for the Python programming language that adds support for opening, manipulating, and saving many different image file formats. When data of images are downloaded, opened image and saved to repository for used Pillow library. Lxml is a Python library which allows for easy handling of XML and HTML files, and can also be used for web scraping. When our dataset is collected with web scraping, the library is worked with collected images on websites. Tensorboard is part of the TensorFlow. Tensorboard is used to analyze model of machine learning algorithms. All experiments are performed on PyCharm version 2020.3.5, Python version 3.7.0 and PC requirements that have Intel (R) Core (TM) i5-7200U CPU @ 2.50GHz 2.71 GHz process unit and NVIDIA GeForce 940MX graphic card in Windows 10 operation system.

5. Experiment Results

In this section, comprehensive experiments are carried out in order to detect forest fire using conventional machine learning algorithms, object detection techniques, deep and hybrid deep learning models. Accuracy (AC), f-measure (FM), precision (PR), mean average precision (mAP), and recall (RC) are employed as evaluation metrics to demonstrate the performance of the models. To calculate evaluation metrics for each model, the number of true-positive (TP), true-negative (TN), false-positive (FP), and false-negative (FN) values are obtained using confusion matrix. The TP refers to the number of operations extracted from the result of the fire by the model for visual data that actually contains the fire in the sample space collected within the scope of the study. TN is the number of visual data that does not actually contain a fire image and is determined by the model to contain no fire. FP refers to the number of visual data that the model classifies as if it were containing fire, although it does not actually contain fire data. FN refers to the number of visual data that is classified as if it does not contain fire data on the model side, even though it actually contains fire data. The ratio of the number of data that a model can classify as false when false and true when true to the total number of data gives the accuracy value given in equation 1.

$$Accuracy = \frac{TP+TN}{TP+NP+TN+FN} \quad (1)$$

Precision gives the ratio of correctly predicted positive observation results to total positive estimates, and the power of negative interpretation gives the ratio of correctly predicted negative observation results to total negative estimates. The formula should be given as in equation 2.

$$Precision = \frac{TP}{TP+NP} \quad (2)$$

Recall is used to measure the precision of the model. It gives the ratio of correctly predicted positive estimates to the number of all images that are actually correct (containing fire) given in equation 3.

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

F-measure is a value obtained as a result of the weighted ratio of positive interpretive strength and sensitivity values. Both false-positive and false-negative values are taken into account when calculating the F-measure. When the distribution of classes is not balanced, it gives more realistic results compared to the total accuracy criterion when measuring the success of the model. The formula should be given as in equation 4.

$$F-measure = \frac{2*Recall*Precision}{Recall+Precision} \quad (4)$$

Intersection over Union (IoU) is defined as the area of the intersection divided by the area of the union of a predicted bounding box (B_p) and a ground-truth box (B_{gt}). Formula is presented in equation 5.

$$IOU = \frac{area(B_p \cap B_{gt})}{area(B_p \cup B_{gt})} \quad (5)$$

To observe how the hyperparameters effect the performance of the system, different combinations are tested such as the number of convolution layer, the number of neurons in the hidden layers and dense layer, The best performance result is determined to construct the proposed model. In the experiments, dataset is divided into 70% as training and remaining is test.

The following abbreviations are used in the tables: AC: Accuracy, FM: F-measure, PR: Precision, RC: Recall, SVM: Support vector machine, RF: Random Forest, CNN: Convolutional neural network, CNN-GRU: Convolution Neural Network-Gated Recurrent Unit, CNN-LSTM: Convolutional neural network-long short-term memory, SSD: Single shot detector, Faster R-CNN: Faster recurrent-convolutional neural network, Avg: Average. The best results are obtained for each dataset in the Table 1 and Table 2 after experiments of hyperparameter tuning. The best performance results are also demonstrated in boldface in all tables. In Table 1, the performance results of all classification models according to evaluation metrics in the first dataset (DS1) are demonstrated. As it is clearly seen that in Table 1, convolutional neural network model outperforms other models with 98.32% of accuracy result.

Table 1. *Classification performance results of each method in terms of evaluation metrics on DS1.*

DS1	**Evaluation Metrics**			
Models	**AC**	**FM**	**PR**	**RC**
SVM	72.75	75.86	78.22	83.26

RF	80.26	81.22	83.17	84.35
CNN	98.32	98.12	98.30	98.34
CNN-GRU	53.08	66.34	34.66	66.21
CNN-LSTM	53.45	30.15	15.24	24.68
Avg.	71.24	70.15	62.18	71.96

It is followed by RF with 80.26%, SVM with 72.24 %, CNN-GRU with 53.45%, and CNN-LSTM with 53.08% of accuracy results. Although the RF is the model with the second-best classification ability, it cannot be said to be competitive due to nearly an 18% accuracy difference between them compared to CNN. As another result of Table 1, hybrid deep learning models exhibit poor performance with approximately 53.08% of accuracy in the image classification task for DS1, both compared to traditional machine learning techniques and CNN as a deep learning model.

In Table 2, the performance results of all classification models according to evaluation metrics in the second dataset (DS2) are demonstrated.

Table 2. *Classification performance results of each method in terms of evaluation metrics on DS2.*

DS2	**Evaluation Metrics**			
Models	**AC**	**FM**	**PR**	**RC**
SVM	75.29	78.48	70.29	89.15
RF	81.45	83.08	78.49	88.66
CNN	99.32	99.32	99.22	99.42
CNN-GRU	55.11	52.26	46.48	57.78
CNN-LSTM	54.73	34.25	17.48	37.78
Avg	73.35	69.52	62.46	74.18

As it is obviously observed that in Table 2, convolutional neural network model outperforms other models with 99.32% of accuracy result. It is followed by RF with 81.45%, SVM with 75.29%, CNN-GRU with 55.11%, and CNN-LSTM with 54.15% of accuracy results. Although the RF is the second-best classification technique, RF is not competitive because of about an 18% accuracy decrement when compared to CNN method. As another result of Table 2, hybrid deep learning techniques present the poorest classification success with approximately 54.15% of accuracy for CNN-LSTM and 55.11% of accuracy for CNN-GRU in DS2, both compared to conventional machine learning techniques and CNN model. The classification performance of models is ordered as: CNN> RF> SVM> CNN-GRU> CNN-LSTM.

When we compare Table 1 and Table 2 in terms of datasets, there is no remarkable change in accuracy for the CNN model when the total number of data increases. On the other hand, the traditional machine learning models exhibit an enhancement in accuracy values of nearly 1% for the RF, about 3% for the SVM model, and almost 2% and roughly 1 for the CNN-GRU and CNN-LSTM models when the number of data increases. In addition, when the average accuracy value is considered in Table 1 and Table 2, an increase of almost 2% is also noticeable. Experiment results demonstrate that the utilization of CNN model for detection of forest fire significantly contributes to classification success of the system. For this reason, experiments are carried out on detailed parameter settings of CNN model as a next step. Table 3 and Table 4 present detailed parameter experiments for DS1 and DS2, respectively. The parameters employed in the experiments are the number convolution layer, the number of neurons and dense layer. In order to ensure the best scores of CNN model, different combinations are tested by varying epoch sizes. Epoch size is arranged as 20 for CNN and 50 for hybrid deep models, 32 batch sizes for CNN and 8 batch sizes for hybrid deep models, the number of convolution layer, the number of neurons contained in each layer, and the number of dense layers. The following abbreviations are used in the Table 3 and Table 4: CNV: The number of convolution layer, ND: The number of neurons contained in each layer, DNS: the number of dense layers.

Table 3. *Classification performance results of each parameter combination in terms of evaluation metrics on DS1.*

CNN	**Evaluation Metrics**			
Parameters	**AC**	**FM**	**PR**	**RC**
2 CNV-32 ND-1 DNS	98.32	98.12	98.30	98.34
2 CONV-32 ND-2 DNS	97.26	90.79	90.79	90.79
2 CNV-64 ND-1 DNS	97.26	97.26	97.26	98.26
2 CNV-64 ND-2 DNS	95.29	85.89	80.45	92.28
3 CNV-32 ND-1 DNS	90.42	95.08	93.47	97.74
3 CNV-32 ND-2 DNS	93.13	94.10	94.15	94.13
3 CNV-64 ND-1 DNS	84.46	95.19	93.73	98.19
3 CNV-64 ND-2 DNS	93.54	95.46	98.31	94.46

In Table 3, the combination of 2 convolution layer, 32 nodes, and 1 dense layer exhibits the best classification success with 98.32% of accuracy. When the number of dense layers and convolution layers are set to 1, and 2, respectively, the only change is observed in number of nodes. When the number of nodes increases from 32 to 64, the performance is decreased nearly 1%. If the number of node and dense layers are adjusted as 32 and, 1, the number of convolutions varies to 3 from 2, which causes approximately 8% decrement in classification accuracy. When the number convolution layers, nodes, and dense layers are set to 3, 64, and 1, the fire detection system exhibits the poorest classification success with 84.15% of accuracy. As a result of Table 3 and Table 4, the raise of number of convolution layer and nodes affect the classification performance of the system, negatively. Finally, CNN experiments are carried out by setting the number of convolution layer, nodes, and dense layer as 2, 32, and 1, respectively for DS1 and DS2. In Table 4, classification results of each parameter combination in terms of evaluation metrics on DS2 are presented. Similar to Table 3, the combination of 2 convolution layer, 32 nodes, and 2 dense layers exhibits the best classification success with 99.32 % of accuracy. Although there is a modification in the number of nodes and hidden layers, no significant increase or decrease in classification performance is observed. However, the increase in the number of convolution layers caused a decrease in classification performance of about 5%.

Table 4. *Classification performance results of each parameter combination in terms of evaluation metrics on DS2.*

CNN	Evaluation Metrics			
Parameters	**AC**	**FM**	**PR**	**RC**
2 CNV-32 ND-1 DNS	99.16	98.60	99.52	98.61
2 CNV-32 ND-2 DNS	99.32	99.32	99.22	99.42
2 CNV-64 ND-1 DNS	98.96	99.41	99.34	99.16
2 CNV-64 ND-2 DNS	99.46	99.42	99.49	99.83
3 CNV-32 ND-1 DNS	96.59	95.19	95.24	96.81
3 CNV-32 ND-2 DNS	94.26	94.49	94.34	93.28
3 CNV-64 ND-1 DNS	95.76	96.64	97.57	95.19
3 CNV-64 ND-2 DNS	97.78	96.15	97.42	97.26

In addition to classification task in order to detect forest fire, object detection techniques are also evaluated. For this purpose, SSD and Faster R-CNN models are performed in the experiments. In Table 5, performance results of object detection techniques are presented in terms of evaluation metrics. The mAP metric compares the ground-truth bounding box to the detected box and returns a score. The results demonstrated in Table 5 are ensured employing 1,605 fire images with 300*300 size. Test data contains 232 fire images and train data has 1,379 fire images. Common Objects in Context dataset (MS COCO), which contains 80k training images ("2014 train") and 40k validation images ("2014 val") is released by Microsoft. There is an associated MS COCO challenge with an evaluation metric, that averages mAP when is used calculate object detection accuracy over different IoU thresholds, from 0.5 to 0.95. This emphasizes a significantly larger emphasis on localization compared to COCO metrics. Every model has a performance number for training dataset. When Faster R-CNN model is trained on the dataset, the result of mAP is lower than 28%. Actually, this means that the Faster R-CNN model gives result of mAP score 13.36% for 0.5 intersection on unit (IOU). The same dataset is also trained on other object model single shot detector (SSD). The experiment result shows that the usage of SSD technique outperforms Faster R-CNN with 22.2% of mAP. Furthermore, precision of SSD model performs better than Faster R-CNN method with 19.26%. Figure 2 demonstrates the fire detection results of SSD model on test images.

Table 5. *Performance results of object detection techniques in terms of evaluation metrics.*

	Evaluation Metrics		
Models	**mAP**	**Precision**	**Recall**
SSD	22.20	19.26	43.00
FasterR-CNN	13.36	14.22	43.22

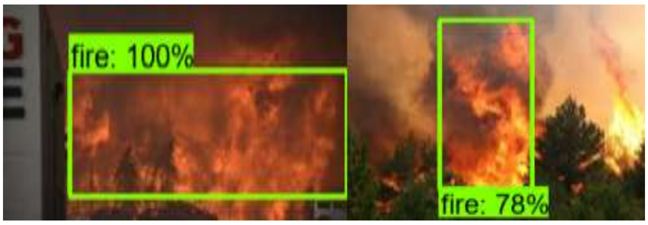

Figure 2. *SSD fire detection model on test images.*

In [25], authors propose multi-scale prediction for fire detection utilizing convolutional neural network. 97.89% of F-measure score is reported while our study demonstrates 98.12% of F-score. In [26], convolutional neural network (CNN) is proposed to detect fire by classifying both fire and smoke in videos. The training procedure is performed with the videos including both fire and smoke. The experiment results indicate that the model used in the study is able to classify the fire, smoke and fire with smoke with a recognition rate of up to 94%, 95% and 93%, respectively. In [27], Saponara et. al present real-time video fire/smoke detection system employing YOLOv2 convolutional neural network. Authors employ a large scale of fire/smoke and negative videos in different environments, both indoor (e.g., a railway carriage, container, bus wagon, or home/office) and outdoor (e.g., storage or parking area). They report that the achieved experimental results show that the proposed system is suitable real-time video-surveillance system for fire/smoke detection with 96.02% of accuracy that surpasses of our object detection models because of using different model and different experiment settings, and datasets.

6. Conclusion and Discussion

Forest fire detection is a very challenging task in the area of object detection. To overcome the problem of fire detection in outdoors, deep learning and conventional machine learning based computer vision techniques are utilized to determine the fire detection when indoor fire detection systems are not capable. In this work, we present a detailed analysis of forest fire detection employing conventional machine learning algorithms, object detection techniques, deep and hybrid deep learning methods. To the best of our knowledge, it is the very first attempt to assess forest fire in such a comprehensive way in terms of six different techniques and two different size data sets. In this aspect, this work contributes to the literature in terms of analyzing various classification and object detection methods in more details that is not addressed before to detect forest fire. Extensive experiments show that convolutional neural networks outperform other techniques with 99.32% of accuracy result. As a future work, we intend to analyze the effects of transfer learning methodology on forest fire detection with large volumed datasets.

Author Statements:

- The authors declare that they have equal right on this paper.
- The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.
- The authors declare that they have nobody or no-company to acknowledge.

References

[1] Den Breejen, E., Breuers, M., Cremer, F., Kemp, R., Roos, M., Schutte, K., De Vries, J. S. Autonomous forest fire detection Coimbra, Portugal: ADAI-Associacao para o Desenvolvimento da Aerodinamica Industrial. pp. 2003-2012, 1998.

[2] Thengade, A., Mishra, P., Kshatriya, R., Mhaskar, R., & Bodhe, P. Fire Detection Using Image Processing Using Raspberry PI.

[3] Kilimci, Z. H., Ganiz, M. C. Evaluation of classification models for language processing. International Symposium on Innovations in Intelligent SysTems and Applications (INISTA) (pp. 1-8). IEEE. 2015

[4] Deng, L., Hinton, G., Kingsbury, B. New types of deep neural network learning for speech recognition and related applications: An overview. In 2013 IEEE international conference on acoustics, speech and signal processing (pp. 8599-8603). IEEE. 2013

[5] Singh, S. P., Kumar, A., Darbari, H., Singh, L., Rastogi, A., Jain, S. Machine translation using deep learning: An overview. In 2017 international conference on computer, communications and electronics (comptelix) (pp. 162-167). IEEE. 2017

[6] Khondaker, A., Khandaker, A., Uddin, J. Computer Vision-based Early Fire Detection Using Enhanced Chromatic Segmentation and Optical Flow Analysis Technique. International Arab Journal of Information Technology, 17(6), 947-953.2020

[7] A. Rafiee, R. Dianat, M. Jamshidi, R. Tavakoli, and S. Abbaspour, Fire and smoke detection using wavelet analysis and disorder characteristics, ICCRD 2011 - 2011 3rd Int. Conf. Comput. Res. Dev., 3 262–265. 2011

[8] Mahmoud, M. A. I., Ren, H. Forest fire detection and identification using image processing and SVM. Journal of Information Processing Systems, 15(1), 159-168. 2019

[9] Sadewa, R. P., Irawan, B., Setianingsih, C. Fire detection using image processing techniques with convolutional neural networks. In 2019 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI) (pp. 290-295). IEEE. 2019

[10] Sucuoğlu, H. S., Böğrekçi, İ., Demircioğlu, P. Real Time Fire Detection Using Faster R-CNN Model. International Journal of 3D Printing Technologies and Digital Industry, 3(3), 220-226. 2019

[11] Mwedzi, N. A., Nwulu, N. I., Gbadamosi, S. L. Machine Learning Applications for Fire Detection in a Residential Building. In 2019 IEEE 6th International Conference on Engineering Technologies and Applied Sciences (ICETAS) (pp. 1-4). IEEE. 2019

[12] Abid, F. A Survey of Machine Learning Algorithms Based Forest Fires Prediction and Detection Systems. Fire Technology, 57(2), 559-590. 2021

[13] Xu, R., Lin, H., Lu, K., Cao, L., Liu, Y. A Forest Fire Detection System Based on Ensemble Learning. Forests, 12(2), 217. 2021

[14] Pourghasemi, H. R., Gayen, A., Lasaponara, R., & Tiefenbacher, J. P. Application of learning vector quantization and different machine learning techniques to assessing forest fire influence factors and spatial modelling. Environmental research, 184, 109321.7. 2020

[15] Maksymiv, O., Rak, T., Peleshko, D. Real-time fire detection method combining AdaBoost, LBP and

convolutional neural network in video sequence. In 2017 14th International Conference the Experience of Designing and Application of CAD Systems in Microelectronics (CADSM) (pp. 351-353). IEEE. 2017

[16] Cortes, C., Vapnik, V. Support-vector networks. Machine learning, 20(3), 273-297. 1995

[17] Azarmdel, H., Jahanbakhshi, A., Mohtasebi, S. S., & Muñoz, A. R. Evaluation of image processing technique as an expert system in mulberry fruit grading based on ripeness level using artificial neural networks (ANNs) and support vector machine (SVM). Postharvest Biology and Technology, 166, 111201. 2020

[18] Ho, T. K. Random decision forests. In Proceedings of 3rd international conference on document analysis and recognition (Vol. 1, pp. 278-282). IEEE. 1995

[19] Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078. 2014

[20] Shashua, A. Introduction to machine learning: Class notes 67577. arXiv preprint arXiv:0904.3664. 2009

[21] https://cs231n.github.io/convolutional-networks/

[22] Nagrath, P., Jain, R., Madan, A., Arora, R., Kataria, P., & Hemanth, J. SSDMNV2: A real time DNN-based face mask detection system using single shot multibox detector and MobileNetV2. Sustainable cities and society, 66, 102692. 2021

[23] Ren, S., He, K., Girshick, R., Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. arXiv preprint arXiv:1506.01497. 2015

[24] Hochreiter, S., Schmidhuber, J. Long short-term memory. Neural computation, 9(8), 1735-1780.

[25] Jeon, M., Choi, H. S., Lee, J., & Kang, M. (2021). Multi-scale prediction for fire detection using convolutional neural network. Fire Technology, 2021, 1-19. 1997

[26] Singh, A. R., Athisayamani, S., Narayanan, S. S., & Dhanasekaran, S. Fire detection by parallel classification of fire and smoke using convolutional neural network. In Computational Vision and Bio-Inspired Computing (pp. 95-105). Springer, Singapore. 2021

[27] Saponara, S., Elhanashi, A., & Gagliardi, A. Real-time video fire/smoke detection based on CNN in antifire surveillance systems. Journal of Real-Time Image Processing, 18(3), 889-900. 2021