ORIGINAL RESEARCH Open Access

Forest fire and smoke detection using deep learning-based learning without forgetting

Veerappampalayam Easwaramoorthy Sathishkumar[1], Jaehyuk Cho[1*], Malliga Subramanian[2] and Obuli Sai Naren[2]

Abstract

Background Forests are an essential natural resource to humankind, providing a myriad of direct and indirect benefits. Natural disasters like forest fires have a major impact on global warming and the continued existence of life on Earth. Automatic identification of forest fires is thus an important field to research in order to minimize disasters. Early fire detection can also help decision-makers plan mitigation methods and extinguishing tactics. This research looks at fire/smoke detection from images using AI-based computer vision techniques. Convolutional Neural Networks (CNN) are a type of Artificial Intelligence (AI) approach that have been shown to outperform state-of-the-art methods in image classification and other computer vision tasks, but their training time can be prohibitive. Further, a pretrained CNN may underperform when there is no sufficient dataset available. To address this issue, transfer learning is exercised on pre-trained models. However, the models may lose their classification abilities on the original datasets when transfer learning is applied. To solve this problem, we use learning without forgetting (LwF), which trains the network with a new task but keeps the network's preexisting abilities intact.

Results In this study, we implement transfer learning on pre-trained models such as VGG16, InceptionV3, and Xception, which allow us to work with a smaller dataset and lessen the computational complexity without degrading accuracy. Of all the models, Xception excelled with 98.72% accuracy. We tested the performance of the proposed models with and without LwF. Without LwF, among all the proposed models, Xception gave an accuracy of 79.23% on a new task (BowFire dataset). While using LwF, Xception gave an accuracy of 91.41% for the BowFire dataset and 96.89% for the original dataset. We find that fine-tuning the new task with LwF performed comparatively well on the original dataset.

Conclusion Based on the experimental findings, it is found that the proposed models outperform the current state-of-the-art methods. We also show that LwF can successfully categorize novel and unseen datasets.

Keywords Forest fire, Image processing, Deep learning, CNN, Learning without forgetting, Transfer learning

*Correspondence:
Jaehyuk Cho
chojh@jbnu.ac.kr
[1] Department of Software Engineering, Jeonbuk National University, Jeonju-si, Jeollabuk-do 54896, Republic of Korea
[2] Department of Computer Science and Engineering, Kongu Engineering College, Erode 638060, Tamil Nadu, India

Resumen

Antecedentes Los bosques son un recurso natural esencial para la humanidad, proveyendo una miríada de beneficios directos e indirectos. Los desastres naturales como los incendios forestales tienen un gran impacto sobre el calentamiento global y la existencia continua de la vida sobre la tierra. La identificación automática de los incendios forestales es entonces un campo importante para la investigación de manera de evitar desastres. La detección temprana de incendios puede también ayudar a los decisores a planificar métodos de mitigación y tácticas de extinción. Esta investigación se enfoca a la detección de fuegos/humos desde imágenes usando técnicas de visión computacional basada en Inteligencia artificial. Las redes neuronales convolucionles (CNN) es un tipo de aproximación de la inteligencia artificial (AI) que ha demostrado superar al estado del arte de los métodos de clasificación de imágenes y otras técnicas de visión mediante computadoras, aunque su tiempo de entrenamiento puede ser prohibitivo. Además, una persona previamente entrenada en CNN puede bajar su performance si el conjunto de datos resulta insuficiente. Para enfocar este problema, la transferencia en el aprendizaje es ejercitado en modelos pre entrenados. Por supuesto, esos modelos pueden perder sus habilidades clasificatorias del conjunto de datos originales cuando se aplica la transferencia de conocimientos. Para resolver este problema, usamos el Conocimiento sin Olvidos (LwF), el cual entrena a la red con una nueva tarea pero mantiene intactas las habilidades preexistentes.

Resultados En este estudio, implementamos la transferencia de aprendizaje en modelos pre entrenados como el VGG16, Inception V3, y Xception, que nos permite trabajar con conjuntos de datos pequeños y simplifican la complejidad computacional sin degradar la exactitud. De todos los modelos, Xception dio una exactitud de 79,23% en una nueva tarea (Bow Fire dataset). Cuando usamos LwF, Xception dio una exactitud de 91,41% para el Bow Fire dataset, y 96,89% para el conjunto de datos original. Encontramos que sintonizando de manera fina la nueva tarea con LwF, éste se comportó comparativamente bien en el conjunto de datos original.

Conclusiones Basados en los resultados experimentales, encontramos que los modelos propuestos tienen mejores resultados que los métodos que representan el actual estado del arte. También mostramos que LwF puede exitosamente caracterizar conjuntos de datos noveles y nunca vistos.

Introduction

Forest fires are a common occurrence worldwide due to climate change, which results in severe economic losses and ecological destruction (Bot 2022; Castelli et al. 2015). Forest fires can be natural or man-made forest fires and summer forest fires caused by debris and other biomes, as well as human negligence. Even though, wildfires can benefit local vegetation, animals, and ecosystems, but they can also cause major damage to property and human life. In recent years, the frequency of forest fire accidents has been continuously increasing. Hence, there has been a rise in interest of implementing systems for automated observation and detection of forest fires, as a means of protecting forests from destruction.

There are a number of conventional and cutting-edge fire and smoke detection techniques that have been proposed to reduce damage brought on by fire disasters. Sensor-based and vision-based smoke detection systems have garnered a lot of interest in the research community among these techniques. Based on sensor types and applications, the fire detection technique is split into five basic groups: smoke-sensitive, light-sensitive, gas-sensitive, temperature-sensitive, and composite (Saeed et al. 2018). Temperature and smoke sensors are frequently used for this purpose (Kizilkaya 2022). The sensor-based approach has significant limitations in terms of detection range and detection speed (Park and Ko 2020). Since fire spreads quickly, it is important to keep the delay as short as possible. Then, as video surveillance technology came up, researchers gathered fire images and used their color characteristics to look for fires. Orange or yellow flames moving side to side are the most common visual representations of fire in videos and images. Soot or burnt particles can be seen in smoke as a blend of white, gray, and black plumes. Smoke detection in videos and images has its own set of difficulties. To be effective, a system must be able to find the difference between images that truly contain the fire and those that appear to have flames but are not. False alarm rates are higher when using simple color features for fire detection (Hu et al. 2022). So, in order to capture the properties of a fire, such as color, shape, flickering, frequency, and dynamic textures, image processing-based methods have been developed. These techniques detect fire by utilizing the RGB, YUV, YCbCr, and CIELab color spaces (Yang 2022; Al-Duryi 2022; Fang 2022; Seydi 2022).

Along with color information, motion data has also been incorporated. The reliance on fire detection technology has grown as a result of methods discussed in (Anh et al. 2022). But, the use of surveillance camera images has introduced a new image processing issue. Video cameras produce a continuous stream of images that must be stored and processed which would be more expensive. As a result, several approaches and systems for fire detection have been presented to make the system as precise and autonomous as feasible. As video surveillance technology expanded in recent years, image processing technology in machine vision also advanced (Zhao et al. 2022), speeding up transmission and sensing. As a result, computer vision-based fire and smoke detection technology has been developed, enabling a greater variety of fire detection approaches. By utilizing video surveillance to collect and extract features from the images of fire and smoke, a computer vision-based fire and smoke detection technology can develop a detection model that relies on these images. Hence, to assess the presence of fire and smoke in images, traditional machine learning and deep learning-based computer vision approaches have been advocated.

Machine learning has been used in a range of applications, including forest fire prediction and detection. (Abid 2021; Arif et al. 2021; Ko et al. 2009; Kong et al. 2016; Bouguettaya et al. 2022; Friggens 2021) provides a wide-ranging overview of the use of machine learning techniques for forest fire detection. Machine learning-based fire detection algorithms rely on manually extracting visible information from images. These characteristics only focus on the shallow characteristics of the flame, which could lead to data loss when extracting manually. Unlike machine learning algorithms, deep learning (Schmidhuber 2015) can automatically extract and learn complicated feature representations. CNN's success in image classification and deep learning's breakthrough growth in computer vision (Ha 2018; Mao et al. 2018; Saeed et al. 2020; Yang et al. 2019; Li et al. 2020; Majid et al. 2022; Fouda 2022) make fire detection a promising area of research. CNN-based methods use frames from surveillance systems as input, and the prediction result is sent to an alert system. Inception (Szegedy 2015), VGG-Net (Simonyan 2014), Xception (Chollet 2017), and many more CNN variants have been applied in fire detection tasks.

Classifying images of fire and smoke has proven difficult in the past due to the large parameter space used by off-the-self deep architectures such as VGG16, DenseNet, Inception, and Xception, among other options. When faced with large parameter spaces, however, transfer learning may be a viable option. Knowledge learned in one domain can then be transferred to another where there is fewer data. Even with a few images, deep architectures using pre-trained models can be built (Best et al. 2020). When trained on a large number of examples, deep learning models outperform (Tian 2015). When training samples are inadequate, overfitting and slipping into a local optimum can occur (Krizhevsky et al. 2017). Transfer learning can aid us in resolving such situations. Many computer vision tasks, such as object detection and face recognition, have seen recent success with deep learning, but the use of these approaches for fire detection has been sparse. Fire detection research may be lacking due to a shortage of data for deep learning models. This has motivated us to focus on the collection of a considerable quantity of fire/smoke images from different sources. Further, even if a pre-trained CNN classifier is trained to classify particular types of tasks using transfer learning, the fact is that the model can work well on recognizing tasks on which it has been trained, but it underperforms when a new, but similar task is given. This is known in machine learning, as "the catastrophic forgetting phenomenon." This phenomenon further motivated us to explore the concept of LwF for detecting forest fire /smoke images from a new dataset. The focus of the proposed research work is highlighted below:

Research focus

Following are the research questions we would like to address in this work.

RQ1: How are pre-trained models adaptable?

To address this question, we compared the performance of pre-trained models as feature extractors and fine tuners.

RQ2: How well do pre-trained models categorize new dataset images?

To address this, we refined and trained numerous pre-trained CNN models and compared them to models employed solely as feature extractors.

RQ3: To what extent may fine-tuning hyperparameters for various CNN models be effective?

Because the choice of values of hyperparameters is critical for evaluating a model's performance, we employed Bayesian Optimization to determine the ideal values for the hyperparameters.

RQ4: Is the knowledge gained by models from one dataset transferable to another?

We used BoWFire, a small yet challenging dataset, to investigate this issue.

The objective of this work is to build a set of models that automatically recognize and detect the presence of fire/smoke in images using pre-trained CNN models like VGG16, InceptionV3, and Xception architectures. By utilizing two techniques namely freezing the convolutional base (feature extractor) and training some convolutional layers while freezing others (fine-tuner), we can use previously learned models for new tasks. Furthermore, we use the LwF, which trains the network using only new task data while preserving its baseline capabilities. Additionally, Bayesian optimization is used in this study to identify the best hyperparameter configuration because it is crucial and challenging to select the right hyperparameters when training CNN architectures.

Research contributions

The contributions to this work include:

(i) Examined various pre-trained CNN models and identified the methods to explore the pre-trained models.
(ii) Developed low-cost computation models and analyzed the performance of the variants of CNNs.
(iii) Optimized the values for various hyperparameters of CNN models using Bayesian optimization.
(iv) Transferred the knowledge learned by the proposed models to a standard, but challenging dataset, BoWFire using LwF.

As far as we are aware, there is not any work in the literature that discusses transfer learning utilizing LWF, fine-tuning procedures, and optimization approach for categorizing fire and smoke images. The remainder of the article is organized as follows: the "Literature survey" section discusses recent developments in the field of fire and smoke detection. In the "Materials and methods" section, we discuss the dataset, deep neural network architectures, tuning of hyperparameters, and fine-tuning procedures. This section also introduces LwF. The "Experiments and results" section presents the experimental results. The "Findings and discussion" section summarizes the findings from the study. In the "Findings and discussion" section, an in-depth look at how the images were wrongly classified by the proposed models is also given. Finally, in the "Conclusion and feature direction" section, we recapitulate our study and give a direction for future works.

Literature survey

This section discusses the many research efforts that have been conducted to build models for detecting fire and smoke detection systems. With the growth of AI, numerous research attempts have been made to detect the presence of fire/smoke in images using machine learning and deep learning models. However, in this work, we examined CNN-based models for fire/smoke detection.

In a range of computer-based vision applications, such as visual recognition and image classification, the introduction of CNNs has resulted in significant performance gains. By recognizing hand-written characters, LeNet-5, a CNN algorithm presented by LeCun et al. (1998), achieved one of the first successful outcomes in this field. Due to the availability of large-scale datasets and the advent of incredibly powerful GPUs, researchers have lately been able to generate extremely deep CNNs. For instance, Krizhevsky et al. (Best et al. 2020) introduced AlexNet, a deep CNN network that performed exceptionally well in the 2012 ImageNet Challenge. Additionally, numerous CNN variations have exhibited exceptional performance in image categorization (Namozov and Im Cho 2018).

CNNs in smoke and fire detection were examined in a survey (Li and Zhao 2020). Further, this effort also discussed current datasets and overviews of modern computer vision approaches. In conclusion, the authors highlighted the obstacles and potential solutions for furthering the development of CNNs in this field. Mahmoud et al. (2022) developed a time-efficient fire detection system using CNN and transfer learning. This model leveraged a CNN architecture with an acceptable computing time for real-time applications and asserted that the proposed model required less training and classification time than existing models in the literature due to the use of transfer learning. Bari et al. (Bari 2021) used their curated v3-base dataset of online and recorded videos to fine-tune the InceptionV3 and MobileNetV2 models. The authors found that when trained on a small dataset, transfer learned models outperform fully trained models. The authors of (Cheng 2021) developed an approach using a Fast Regional Convolutional Neural Network (Fast R–CNN). A selective search method was used to locate candidate images from the sample images. As proven by the results, fast R-CNN smoke detection showed an increased detection rate and decreased false alarms. Pu and Zhao (Li and Zhao 2020) proposed novel fire detection methods based on advanced object identification CNN models such as Faster-RCNN, R–FCN, YOLO v3 etc. A comparison of proposed and existing fire detection algorithms indicated that those based on object detection, CNNs outperformed other algorithms in terms of accuracy. And, YOLOv3-based model gave an average precision of 83.7%, which is much greater than the precision of the other proposed algorithms.

Sousa et al. (2020) summarized recent research attempts to present the common challenges and limitations of these approaches, as well as issues about the dataset quality. Furthermore, they devised a method for transfer learning and utilizing data augmentation techniques that were validated using a tenfold cross-validation scheme. The proposed framework enabled the use of an open-source dataset containing images from over 35 real-world fire events. Unlike video-based works, this dataset contains a high degree of variation between samples, allowing us to test the method in a variety of real-world scenarios. Fernandez et al. (2021) demonstrated a system that can acquire real-time images and process them to perform object detection tasks using RetinaNet and Faster-RCNN. To help contain wildfires, this system is capable of detecting smoke plumes over a large area and communicating with and alerting authorities. Luo et al. (2018) developed a smoke detection system using a CNN and the motion characteristics of smoke. To begin, they identified candidate regions using a combination of the background dynamic update and a priori dark channel technique. Following that, using a CNN, the candidate region's features were extracted automatically.

With the use of optical images and retrained VGG16 and ResNet50 models, the authors of (Sharma 2017) were able to distinguish between images that included and did not contain the fire. It's worth noting that they created an unbalanced training dataset that included a higher proportion of non-fire images. For fire detection and disaster management, the authors of (Muhammad et al. 2018) integrated AlexNet as a foundation architecture. This system incorporated an adaptive priority mechanism for surveillance cameras, enabling high-resolution cameras to be activated to confirm the fire and assess the data in real time. Inspired by GoogleNet architecture, Muhammad et al. (2018) proposed a fine-tuned CNN model for fire detection in surveillance systems. The tests demonstrated that the suggested architecture outperformed both existing hand-crafted feature-based and AlexNet-based fire detection systems. The authors of (Nguyen et al. 2021) suggested a unique approach for fire detection based on the use of CNN to extract both spatial and temporal information for fire classification from video image sequences. The system extracted image features using a CNN network and then classified them using short- and long-term stages. Experiments on readily accessible public datasets indicated encouraging performance outcomes when compared to prior studies.

Qin et al. (2021) suggested a system for detecting and locating the firing position in images using a depth-wise separable CNN and YOLOv3. To begin, fire images have been classified using a depth-wise separable CNN, which greatly reduces detection time while retaining detection accuracy. Second, YOLOv3 is utilized to locate the position of fire from the images labeled as fire, thus avoiding the problem of detection accuracy being degraded when YOLOv3 is used. Simultaneously, for images without fire, the detection time for target regression is greatly lowered. Validated against a publicly available network database, the tests obtained a detection precision of approximately 98%. In the work by (Jeon et al. 2021), the authors developed a framework for multi-scale prediction using the feature maps created by densely stacked convolutional layers. This approach presented a feature-squeeze block as a mechanism for incorporating feature maps with varying scales into the final forecast. The feature-squeeze block efficiently utilized the multi-scale prediction information by spatially and channel-wise compressing the feature maps. The suggested strategy outperformed currently available CNN-based methods in experiments.

A CNN-based fire detection system appropriate for power-constrained devices was proposed by Vinicius et al. (de 2022). To decrease the computational cost of a deep detection network while attempting to maintain its original performance, this method involves training the network and then eliminating its less crucial convolutional filters. Dampage (2022) presented a system and technique for using a wireless sensor network to identify forest fires in their earliest stages. In addition, for more precise fire detection, a machine learning regression model is proposed. In their work, Dogan et. al.(2022) suggested deep learning models using ResNet and InceptionNet to detect fire from images. These models have been used for extracting the features and these features have been classified using SVM. The authors demonstrated that ResNet gave better performance.

From the above review works, it is clear that CNNs offer tremendous promise for fire detection and can aid in the creation of a robust system capable of significantly reducing human and financial loss due to fires. From the investigation of the literature, we find that even though, the detection of forest fire/smoke from images has been focused on, no work has focused on the forgetting phenomenon when the trained models are used for new tasks of fire/smoke images. Additionally, some gaps in the application of CNN for fire and smoke detection remain including faster training, parameter efficiency, hyperparameter tuning, and transfer learning over the new datasets. Although a few experiments employed transfer learning to expedite the training process, none of the studies mentioned above attempted to tune hyperparameters. To recap, we create a few classification models that can differentiate between fire and smoke in images by combining deep learning and transfer learning with hyperparameter tuning, reducing time and ensuring early detection. In addition, we employ LwF to keep the original network capabilities while training the models on a new data.

Materials and methods

Dataset description and augmentation

Geostationary weather satellites including MODIS, VIIRS, Copernicus Sentinel-2, and Landsat-8 were used to construct the dataset for the proposed study (Kaulage 2022). These satellites are used for fire detection all around the world due to their excellent temporal precision and ability to detect fires in far-off locations. In addition to images from Google and Kaggle (https://www.kaggle.com/datasets/phylake1337/fire-dataset, http://github.com/aiformankind/wildfire-smoke-dataset, http://www.kaggle.com/datasets/dataclusterlabs/fire-and-smoke-dataset), satellite imagery of the forest fire has also been compiled. Manual labeling has been applied to the images, designating them as Fire, No Fire, Smoke, and Smoke Fire. There are 4800 images in the obtained dataset. To expand the number of images, image augmentation techniques such as shifting, flipping, rotating, scaling, blurring, padding, cropping, translation, and affine modification were applied. The collection comprises 6,911 images after augmentation. Then, the datasets for training, validation, and testing were divided, with 80% of the dataset going toward training the classifier and 10% going toward testing and validation. The distribution of images in the dataset for training, testing, and validation is shown in Table 1. Sample images from the dataset are shown in Fig. 1.

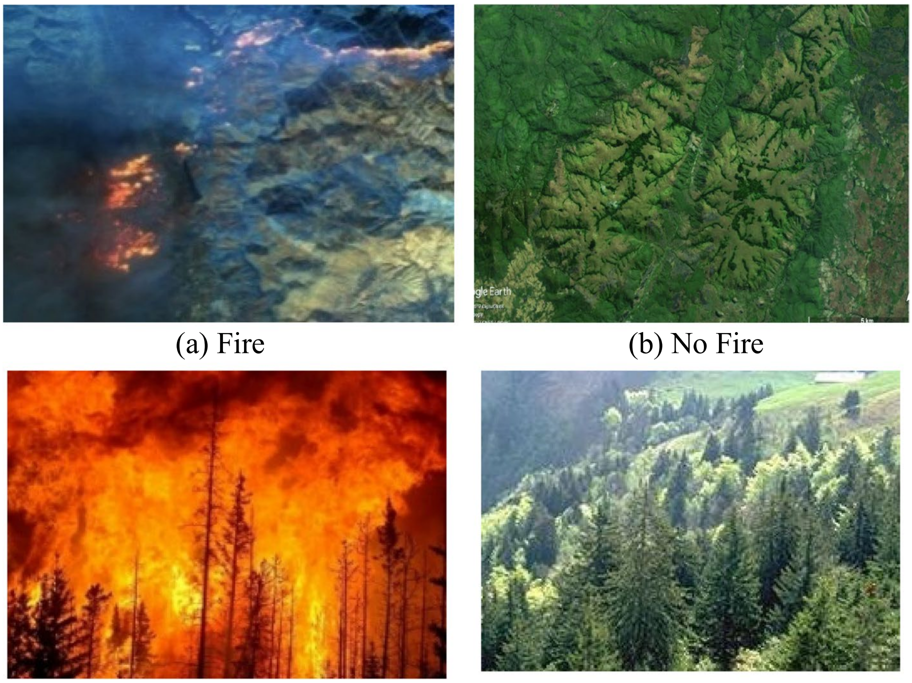
Table 1 Dataset description

Dataset	Fire	No Fire	Smoke	Smoke Fire	Total
Train	2161	2150	490	510	5311
Validation	200	200	200	200	800
Test	200	200	200	200	800

In addition to the compiled dataset, we have used the BoWFire dataset to assess how well the suggested models transfer the knowledge gained from classifying forest fire and smoke images. The BoWFire dataset (http://bitbucket.org/gbdi/bowfire-dataset/downloads/) contains 240 images divided into four categories: fire images, no-fire images, smoke fire and smoke images. Despite its tiny size, this dataset presents significant challenges due to the presence of fire-like sunset and sunrise situations, fire-colored objects, and architectural lighting. A sample from each class is shown in Fig. 2.

CNN variants

Complex vision issues have been efficiently solved using a variety of CNN basic architectures. Convolution and pooling are the two fundamental operations in CNN. The ability to extract features from the images using the convolution operation with various filters allows for the preservation of the corresponding spatial information. Reducing the dimensionality of feature maps produced by the convolution operation using the pooling technique is known as subsampling. The two most popular pooling methods utilized by CNN are max pooling and average pooling. CNNs are utilized as feature extractors and classifiers in image processing applications, notwithstanding their utility in image processing and classification. Rather



(a) Fire

(b) No Fire

(c) Smoke

(d) Smoke Fire

Fig. 1 Sample images from each class in dataset

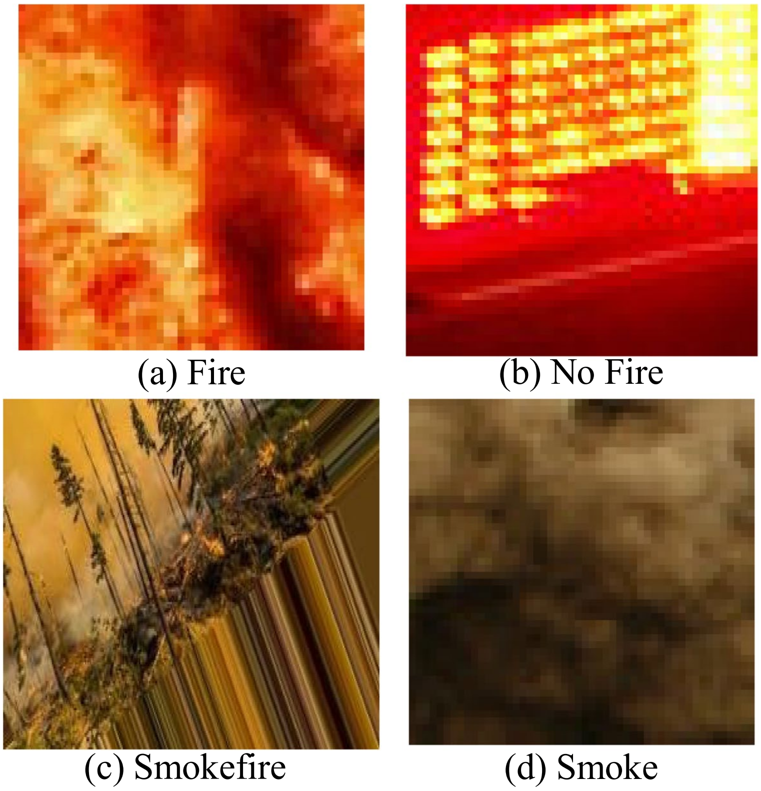


Fig. 2 Sample images from BoWFire dataset

than relying solely on stacked convolutional layers like LeNet, AlexNet, and VGG, current network designs like ResNet, Inception, and Xception are exploring innovative approaches to create convolutional layers to enhance learning efficiency. VGG is a typical CNN architecture, yet it's extensively used because of its simplicity. In this study, we train VGG16, InceptionV3, and Xception to classify fire images. The following section discusses the pre-trained models that have been employed in this study.

VGG16

VGG16 (Visual Geometry Group) is extensively used CNN architecture that is utilized in ImageNet, a big visual database project. VGG16 is widely utilized in a variety of deep learning image classification approaches due to its simplicity of implementation. Despite its 2014 introduction, it remains one of the greatest vision architectures to date. Without altering the receptive fields, VGG uses 1×1 convolutional layers to make the decision function less linear. VGG can have a lot of weight layers because the convolution filters are tiny; of course, having more layers results in better performance.

InceptionV3

Inception-v3 is a CNN architecture derived from the Inception family by several changes such as smoothed-label, batch normalization etc. InceptionV3 focuses mostly on using less computing power by changing the previous Inception architectures so that they are more efficient. It has been found that, in comparison to VGG-Net, Inception networks are more computationally efficient. As a result of this efficiency, Inception networks create fewer parameters and utilize fewer resources than their predecessors. To make InceptionV3 work better for the project, we used factorized convolutions, regularization, dimension reduction, and parallel computations to make the network more efficient.

Xception

Separable Convolutions are replaced with depth-wise Separable Convolutions in the Xception Architecture. Xception is an "extreme" variation of an Inception module. Xception outperforms InceptionV3 on the ImageNet dataset and significantly excels it on a larger dataset with 17,000 classes. Depthwise Separable convolutions will require only less computation compared to separable convolutions. Hence, Xception requires less number of parameters compared to other CNN variants. On the downside, depth-wise 2D convolutions can actually be slower than standard 2D convolutions, although using less memory. Importantly, it has the same number of model parameters as Inception, which results in greater computing efficiency. Xception and Inception vary in yet another way. After the first operation, the existence or absence of non-linearity. Non-linearity is introduced in the Inception model by filtering and compressing input space, but Xception does not.

VGG16, which was developed as a deep CNN, surpasses ImageNet on multiple tasks and datasets. It is intended to reduce the number of parameters in convolution layers and accelerate training time. This makes VGG16 one of the most popular models for image recognition. InceptionV3 cuts processing costs dramatically while retaining speed and precision. In InceptionV3, directed acyclic graphs allow powerful processing. The Xception architecture outperformed VGG16, ResNet, and InceptionV3 on the ImageNet dataset and in the majority of classical classification problems. Traditional network architectures, such as VGG16, are exclusively built of stacked convolutional layers, but newer network architectures, such as InceptionV3 and Xception, seek novel and innovative ways to construct convolutional layers in order to increase learning efficiency. Therefore, in this study, the VGG16, InceptionV3, and Xception CNN architectures have been utilized.

Transfer learning

Transfer learning is a machine learning method that involves applying knowledge from a source domain (for example, ImageNet) to a target domain with significantly fewer samples. In practice, this typically entails initializing a model with pre-trained weights from VGGNet, Inception, or another source of pre-trained weights and then either using it as a feature extractor or fine-tuning the final few layers on a new dataset. Transfer learning enables us to repurpose these models for any relevant task, from object identification for self-driving vehicles to caption generation for video clips. We customize a pre-trained model as a feature extractor and fine-tuner in this work, and a brief note on customization follows:

Feature extractor

This method uses previously learned representations to extract meaningful features from new samples. On top of the pre-trained model, we constructed a new classifier to reuse the feature mappings extracted by the previous dataset (ImageNet). It is not necessary to retrain the entire model in this method. The fundamental convolutional network already has features that can be used to identify images in general. However, the pre-trained model's final classification layer is specific to the ImageNet dataset, but we have layers that are specific to the set of classes on which the model is retrained.

Fine-tuner

In this technique, we unfreeze a few top layers of the models and train both the newly added classification and unfrozen layers of the models. The higher-order feature representations of the underlying model can be "fine-tuned" in this way to make them more relevant to the dataset under consideration. In addition to the classification layers, the weights of a few top layers of the convolution base will be retrained throughout the process of fine-tuning. Due to the fact that the early convolution layers learn extremely generic characteristics, as we ascend the network, the layers tend to learn increasingly task-specific features. Consequently, for fine-tuning, the early layers are maintained frozen while the upper layers are retrained. Applying fine-tuning enables us to use pre-trained networks to distinguish classes in untrained datasets. Since the weights of the uppermost layers are retrained on a new dataset, fine-tuning will result in more accuracy than feature extraction-based transfer learning.

Learning without forgetting

The fact that the shared parameters do not effectively reflect discriminative features of the new tasks indicates that feature extraction performs poorly when applied to the new task in most cases. Fine-tuning reduces performance for previous tasks because shared parameters are changed without changing task-specific parameters. Retraining a model on a new dataset may result in the loss of original task-specific parameters and an inability to perform well on the original tasks. This problem has been addressed by implementing the LwF concept proposed by (https://www.kaggle.com/datasets/phylake1337/fire-dataset) and LwF trains the network with new images while keeping its previous capabilities. With this strategy, the old network's capabilities are preserved while the samples from the new task are used to optimize the accuracy of the new task. However, the old task's images and labels are unnecessary for this approach. We used 240 images from the BoWFire dataset to test this method. The LwF procedure used for the proposed work is given below:

1. *Variables*

   Shared parameters → P_S (network parameters updated for original forest fire dataset)

Task-specific parameters for original forest fire dataset → P_O

Task-specific parameters for BowFire dataset → P_n

(X_n, Y_n) ◊ training data and class label for the BowFire dataset

2. *Procedure*

    1. Y_o =Pre-trained CNN(X_n, P_S, P_O) → find Y_o for each image in the BowFire dataset.
    2. Add nodes in the output layer for each class in the BowFire dataset.
    3. Initialize P_n with random weights.

    Train the network with BowFire dataset images.

    4. Compute $\widehat{Yo}$ =Pre-trainedCNN(X_n, $\widehat{Ps}$,$\widehat{Po}$)
    5. Compute $\widehat{Yn}$ =Pre-trainedCNN(X_n, $\widehat{Ps}$,$\widehat{Pn}$)
    6. Compute loss functions for images in the original and BowFire dataset and update $P_{S,}P_O$, and P_n.
    7. Repeat from step 4 till convergence

From the above steps, it can be seen that goal of LwF is to make a model learn new capabilities while keeping its old capabilities working well, without using training data from the old tasks. Figure 3 shows how the proposed models would be used.

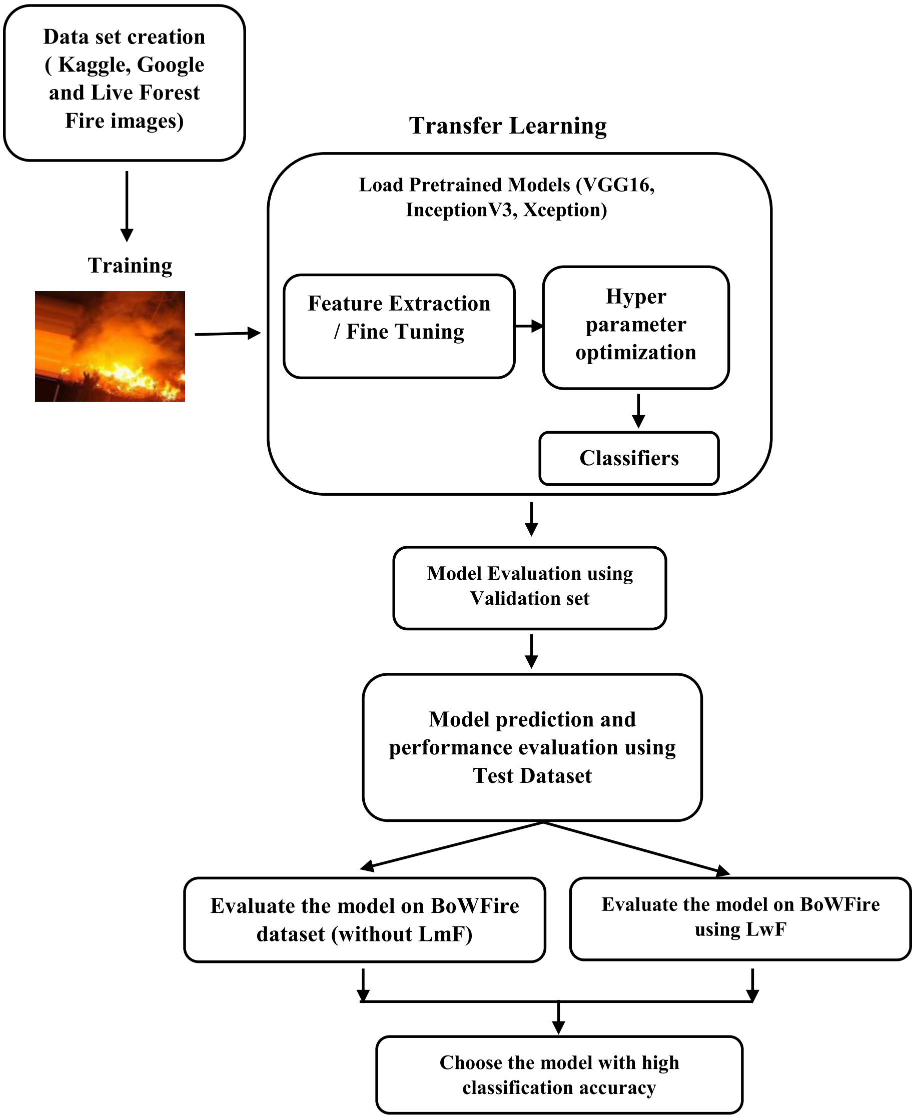

Data set creation (Kaggle, Google and Live Forest Fire images)
Training
Transfer Learning
Load Pretrained Models (VGG16, InceptionV3, Xception)
Feature Extraction / Fine Tuning
Hyper parameter optimization
Classifiers
Model Evaluation using Validation set
Model prediction and performance evaluation using Test Dataset
Evaluate the model on BoWFire dataset (without LmF)
Evaluate the model on BoWFire using LwF
Choose the model with high classification accuracy

Fig. 3 Proposed workflow

Optimization of hyperparameters

Choosing the appropriate hyperparameters for deep learning models is critical for maximizing their potential. A more objective way to do it would be to search for different hyperparameter values and pick the subset that works best on a given dataset. This is referred to as hyperparameter optimization or tuning. The first step in any optimization procedure is defining the search space. The simplest and most frequently used methods for searching are Bayesian optimization, random search, and grid search. In this work, we use Bayesian optimization to choose ideal values for hyperparameters and it runs the models multiple times with different sets of hyperparameter values, but it evaluates previous model information to choose the values for hyperparameters for the newer model. Bayesian Optimization method is said to take less time than other methods to reach the models with the highest accuracy. Hence, we used this search technique for finding the optimal values for the hyperparameters. From the literature survey, we find that learning rate, optimizer, activation function, batch size, number of epochs, and number of neurons have been tuned in many research attempts. Hence, in the proposed work, the above hyperparameters have been optimized using Bayesian Optimization. Table 2 highlights the tuned hyperparameters and their respective search spaces. The tuned values of hyperparameters for different models are presented in Table 3.

Table 3 Hyperparameters with tuned values

Hyperparameters	VGG16	InceptionV3	Xception
Optimizer	Adam	Adam	Adam
Learning rate	1e−01	1e−05	1e−03
Activation function	Elu	Relu	Relu
Number of neurons in customized layers	512	256	256
Number of epochs	100	70	75
Batch size	128	64	64

Experiments and results

We designed experiments to evaluate the performance of the pre-trained models based on feature extraction, fine-tuning, and learning without forgetting. Since the proposed models are deeper, we have used GPU-enabled kernels from Kaggle to train them. Tensorflow and Keras frameworks are used for training the models. The models have been trained using the hyperparameters presented in Table 2. Table 3 shows the tuned values of hyperparameters that generated the best results during training. The models were run for 100 epochs, but we stopped them early. Early stopping is a method in which the model is trained for an arbitrary number of epochs and then stopped when there is no improvement in validation accuracy or reduction in validation loss. As mentioned earlier, we did two different sets of experiments. We took out the classifier from these models and added our own classifier so we could do these experiments. We added two fully connected layers and a softmax layer to VGG16. One fully connected layer and one softmax layer have been added to InceptionV3 and Xception. While fine-tuning the models, we have retrained 5, 8, and 7 top layers of VGG16, InceptionV3, and Xception respectively.

We compared the models to the test data to find out how well they worked. Table 4 gives a summary of how well all of the proposed models have been tested and validated.

As each image in the dataset must be classified into one of the four classes, we evaluated the performance of each model against each of the four classes using accuracy, precision, recall, and F1-score. To calculate TP, FP, TP, and TN, Eqs. (1) to (4) have been used.

$$tp_i = c_{ii} \tag{1}$$

$$fp_i = \sum\nolimits_{l=1}^{n} c_{li} - tp_i \tag{2}$$

$$fn_i = \sum\nolimits_{l=1}^{n} c_{il} - tp_i \tag{3}$$

Table 2 Hyperparameters and their search space

Parameter	Search space
Optimizer	Adam, RMSProp, SGD, Adagrad, Adadelta
Number of neurons in customized layers	64, 128, 256, 512, 1024
Activation function	Relu, Elu, LeakyRelu and Tanh
Learning rate	1e−3, 1e−4, 1e−5, 1e−6
Number of epochs	100, 125, 150, 200
Batch size	32, 64, 128

Table 4 Validation and testing accuracy of the proposed models

Models		Validation accuracy (%)	Testing accuracy (%)
VGG16	Feature extractor	95.18	94.38
	Fine tuner	96.32	95.46
InceptionV3	Feature extractor	93.93	92.04
	Fine tuner	97.87	97.01
Xception	Feature extractor	98.27	97.77
	Fine tuner	99.12	98.72

$$tn_i = \sum_{l=1}^{n} \sum_{k=1}^{n} c_{lk} - tp_i - fp_i - fn_i \tag{4}$$

Accuracy, precision, recall, and F1-score are then calculated as given in Eqs. (5) to (8).

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \tag{5}$$

$$Recall = \frac{TP}{TP + FN} \tag{6}$$

$$Precision = \frac{TP}{TP + FP} \tag{7}$$

$$F1score = \frac{(2 * precision * recall)}{(precision + recall)} \tag{8}$$

With Eqs. (1) to (4), we ran each model and computed TP, FP, FN, and TN using the confusion matrices. A confusion matrix is a visual representation of how closely the prediction results match the actual values. The confusion matrices obtained during model training are shown in Fig. 4. The confusion matrix's diagonal elements stand in for the proper classification. The others, on the other hand, are incorrectly categorized. Predicted classes are shown on the *X*-axis, while actual classes are shown on the *Y*-axis. For example, VGG16 detected six images of type No Fire as Fire, three images of type Smoke as No Fire, and so on. Then, by using TP, FP, FN, and TN, the metrics such as accuracy, precision, recall, and F1-score have been calculated for each of the classes for all the proposed models and presented in Tables 5, 6, and 7.

In addition, we compared the performance of the proposed models to that of recent deep learning models. But the datasets used by these models are not the same as the dataset that we compiled.

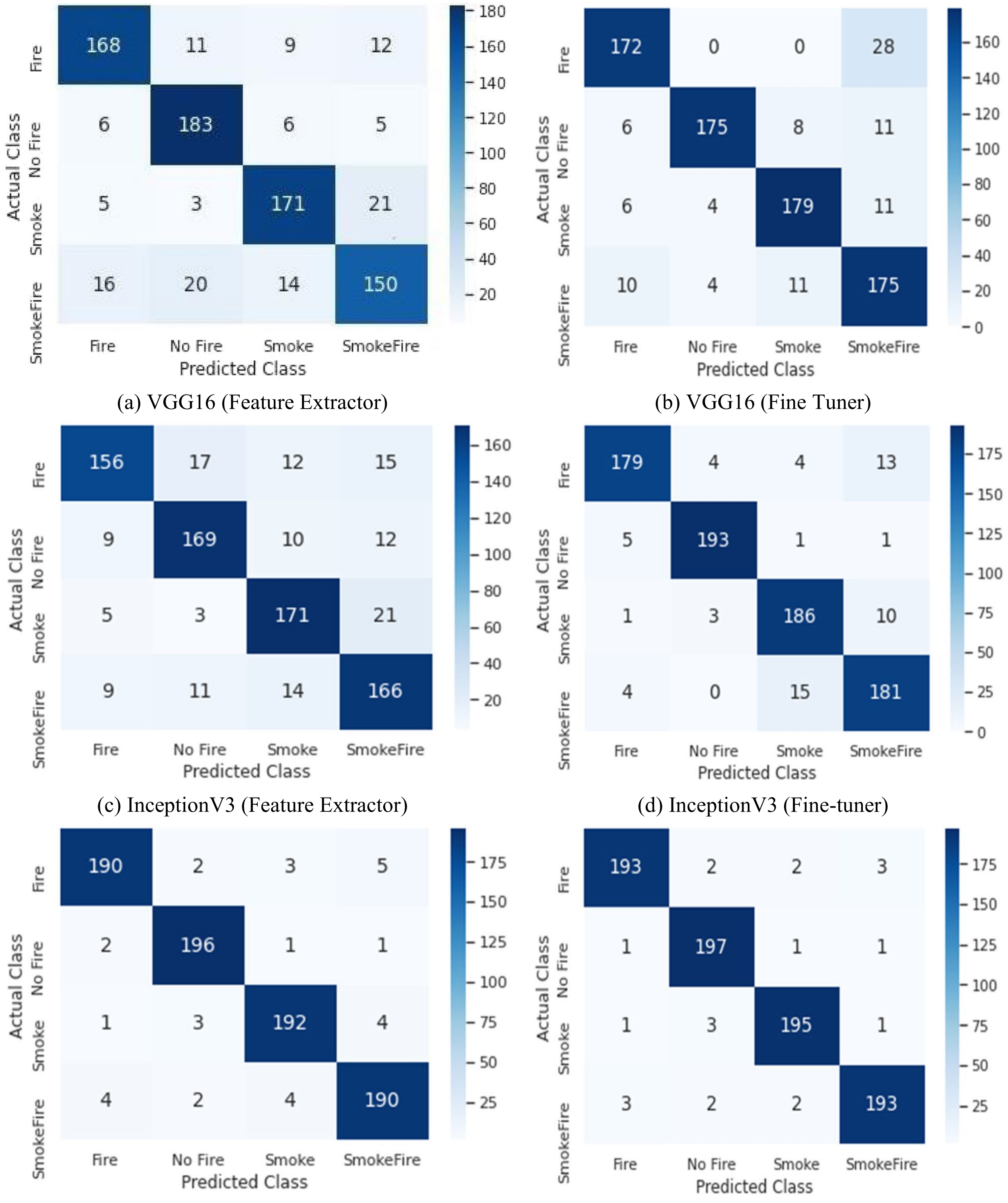
Transfer learning over BoWFire dataset using LwF

Now, we compare the performance of the LwF to that of previously proposed models on the BoWFire dataset. To train the network, LwF only uses new task data, retaining the network's original capabilities. While integrating LwF with the proposed models, the shared parameters (P_S) of the feature extraction layers and task-specific parameters (P_O) of the classification layers for the original dataset (used for training) are retained and the task-specific parameters of BoWFire (P_N) dataset have been updated. Such models learned the parameters that work well on both datasets. For this training, we have used only the images from the BoWFire dataset, that is, the retraining has been done without using the original dataset. To retrain the models on the BoWFire dataset, we have added neurons to the output layer, that is softmax layer, and initialized the weights randomly. The number of newly added parameters is the number of newly added output neurons multiplied by the last shared layer's neurons. This is a small portion of the network's parameters. The procedure for training is enumerated in the "Learning without forgetting" section. To evaluate the performance of LwF, we first tested the pre-trained models on the BoWFire dataset without LwF and then with LwF. The results are shown in Tables 8 and 9.

Findings and discussion

In this work, we intend to classify forest fire images into four classes using traditional and contemporary CNN models. The results of the experiments have been presented in the "Experiments and results" section. In this section, we provide the findings of our research work. To measure the effectiveness of transfer learning techniques used in this work, a set of indicators have been used: the first one being measuring whether the proposed models classify the forest fire dataset using only the transferred knowledge (Feature extractor) and then, finetuning a few top layers of the pre-trained models to find whether there is an increase in the classification accuracy (Finetuner). Since the weights of the pre-trained models have been used as such during feature extraction, the accuracy of the developed models is comparatively less, whereas during finetuning, the weights of a few top layers have been retrained. As a result, the models are better able to learn the features unique to the dataset. Transfer learning works best when employing the network's visual data-trained knowledge on new or related tasks. This reduces training time and improves model accuracy. So, we got better results than other models in the literature and came to the conclusion that models based on transfer learning are better for evaluating classification problems. Among all the proposed models, the Xception model showed the highest performance. One reason for this is that Xception employs depth-wise separable convolution, which facilitates faster and more accurate learning.

While testing the effectiveness of the knowledge transfer of the proposed models over the BoWFire dataset, we find that the accuracy of the models is not appreciable. This is because the models have not been retrained on the BoWFire dataset. And, when we retested the models again on the original forest fire test dataset, the accuracy is not the same as before. It becomes impossible to store and retrain such data as the number of tasks increases. Adding new capabilities to a CNN wipes out the training data for the existing capabilities. As a result, we turned to LwF, which retrains the network using the new task data while preserving the network's original capabilities. According to the results of the experiments, LwF



(a) VGG16 (Feature Extractor)

(b) VGG16 (Fine Tuner)

(c) InceptionV3 (Feature Extractor)

(d) InceptionV3 (Fine-tuner)

(e) Xception (Feature Extractor)

(f) Xception (Fine Tuner)

Fig. 4 Confusion matrix for the proposed models

Table 5 Performance of VGG16

Types of images	Accuracy (%)		Precision (%)		Recall (%)		F1 score (%)	
	FE	FT	FE	FT	FE	FT	FE	FT
Fire	92.63	93.75	86.15	88.66	84	86	85.06	87.31
Smoke	93.63	95.88	84.33	95.63	91.5	87.5	87.77	91.38
Smoke fire	92.75	95	85.5	90.4	85.5	89.5	85.5	89.95
No fire	89	90.63	79.79	77.78	75	87.5	77.32	82.35

Table 6 Performance of InceptionV3

Types of images	Accuracy (%)		Precision (%)		Recall (%)		F1 score (%)	
	FE	FT	FE	FT	FE	FT	FE	FT
Fire	91.63	96.13	87.15	94.71	78	89.5	82.32	92.03
Smoke	92.25	98.25	84.5	96.5	84.5	96.5	84.5	96.5
Smoke fire	91.88	95.75	82.61	90.29	85.5	93	84.03	91.63
No fire	89.75	94.63	77.57	88.29	83	90.5	80.19	89.38

Table 7 Performance of Xception

Types of images	Accuracy (%)		Precision (%)		Recall (%)		F1 score (%)	
	FE	FT	FE	FT	FE	FT	FE	FT
Fire	97.88	98.5	96.45	97.47	95	96.5	95.72	96.98
Smoke	98.63	98.75	96.55	96.57	98	98.5	97.27	97.52
Smoke Fire	98	98.75	96	97.5	96	97.5	96	97.5
No Fire	97.5	98.5	95	97.47	95	96.5	95	96.98

Table 8 Performance of proposed models on BoWFire and Original Forest Fire Dataset without LwF

Models		Accuracy (%)	
		BoWFire	Original Forest Fire
VGG16	**FE**	51.81	94.38
	FT	55.37	95.46
InceptionV3	**FE**	72.04	92.04
	FT	75.31	97.01
Xception	**FE**	77.14	97.77
	FT	79.23	98.72

Table 9 Performance of proposed models on BoWFire and Original Forest Fire Dataset using LwF

Models		Accuracy (%)	
		BoWFire	Original Forest Fire
VGG16	**FE**	76.32	92.71
	FT	78.47	94.46
InceptionV3	**FE**	87.12	93.82
	FT	89.26	95.64
Xception	**FE**	89.41	94.66
	FT	91.41	96.89

outperforms commonly used fine-tuning adaption techniques on the BoWFire dataset and comparatively well on the original dataset. For increased performance on new tasks, LwF may be possible to replace fine-tuning using similar old and new task datasets. As a result, we may deduce that the accuracy of the old task will be equivalent to that of the original network provided the model is maintained in such a way that task-specific characteristics from previous datasets give identical outputs on all relevant images.

We have listed a number of research issues that the planned effort would attempt to answer in the "Introduction" section. Now, we briefly discuss how the suggested models have responded to these questions. To examine the adaptability of pretrained models for the classification of forest fire/smoke dataset, we have used these models as feature extractor and fine-tuner. We refined and trained numerous pre-trained CNN models and compared them to models employed solely as feature extractors and the results have been presented in Tables 4, 5, 6, and 7. As can be seen from these tables, the results very well support our research hypothesis. Instead of choosing the values of hyperparameters at random and measuring the performance, we chose the hyperparameters using Bayesian Optimization search algorithm. This optimization helped us to get optimal values for the hyperparameters that yield better results. Further, to validate the performance of the fine-tuned models for a challenging dataset, we retrained these models using LwF on BowFire dataset and the results have been shown in Tables 8 and 9. From these tables, it can be understood that LwF provides good accuracy for both old and new tasks.

Besides, while training the proposed models, we find that the imbalanced dataset introduced unique challenges to the learning process. Rather of using data-level strategies like resampling, we modified the learning process so that the relevance of the smaller classes is raised throughout training time. This is accomplished by giving the loss function class weights. During model training, a total loss for each batch is determined, and the model parameters are then repeatedly modified to minimize this loss. The loss is the total of the errors between the actual and predicted values for all samples. The total is transformed to a weighted sum with class weighting such that each sample contributes proportionally to the loss depending on its class weight. This has solved the imbalance nature of the datasets. From Table 10, it is understood that the proposed work outperforms other methods in the literature. We believe that the finetuning and hyperparameter optimization approach led to the good results. Such attempts were missing in the other methods.

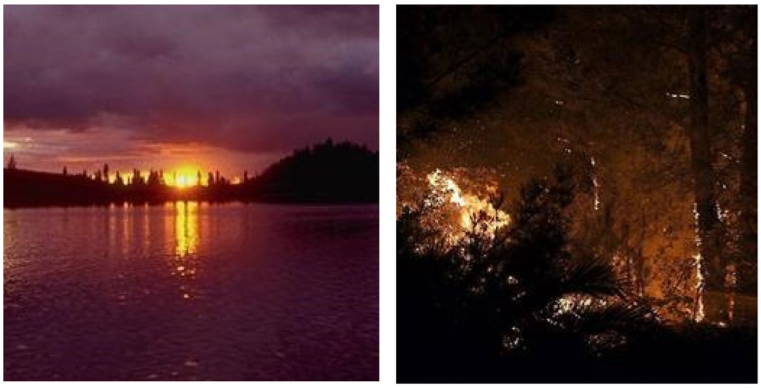
Error analysis

To gain a better understanding of the difficulties inherent in this transfer learning process, we also looked at errors brought on by the suggested models. The process of evaluating test set images that the models incorrectly categorized in order to identify the main reasons for the errors is referred to as "Error Analysis". True positives, false positives, true negatives, and false negatives are all categories for the outcomes of the classification models on images. For example, in the VGG16 model's confusion matrix, we can see that the true positive for the Fire class is 168. This means that out of 200 Fire image samples, 168 have been categorized as Fire and 32 have not. Similarly, only 183 occurrences of the No Fire class have been accurately categorized as No Fire, while 17 instances have been incorrectly classed as not No Fire. We explore a few examples below.

Although the image in Fig. 5a is actually classified as No Fire, the VGG16 model predicted that it is classified as Fire. This is because the values of sunlight pixels are quite near to those of fire color intensities, despite the fact that they are not genuine fire. Although the feature map indicates the existence of sun-rise or set symptoms, we cannot be certain that this is the cause of the misclassification. However, other models have accurately classified the image. Similarly, the VGG16 model predicts the Fire class image shown in Fig. 5(b) as a Smoke class. The VGG16 model's failure to successfully extract the features from the images could be one reason. As a result, reliable forest fire detection

Table 10 Comparative analysis of proposed models with respect to different deep learning models

References	Models	Accuracy (%)
(Li and Zhao 2020)	YOLO v3	83.7
(Mahmoud 2022)	Deep ANN and AlexNet	95 and 98
(Cheng 2021)	VGG16 with TL	97.83
(Guede-Fernández et al. 2021)	Faster R-CNN	80
(Luo et al. 2018)	CNN	90
(Muhammad et al. 2018)	CNN	94.39
(Jeon et al. 2021)	CNN with feature-squeeze block	97.89
Proposed models	VGG16 – Feature Extractor	94.38
	VGG16 – Fine Tuner	95.46
	InceptionV3 – Feature Extractor	92.04
	InceptionV3 – Fine Tuner	97.01
	Xception – Feature Extractor	97.77
	Xception – Fine Tuner	**98.72**



(a) No Fire Image (b) Fire Image

Fig. 5 Error analysis

algorithms continue to be a challenge, because certain objects share characteristics with fire, potentially resulting in a high false alarm rate. The details underlying the misclassification must therefore be ascertained. As a result, we think that making use of the incorrectly categorized images can assist to increase classification precision. Assume that images are frequently mistakenly categorized as belonging to multiple classes when in reality they only belong to one. To gather useful data, we should concentrate on a few misclassified classes rather than analyzing all classes of images.

Furthermore, as compared to still image-based methods, video-based methods successfully enhance fire detection accuracy by lowering both false detections and misclassifications. Such approaches might be particularly effective in distinguishing flames from fire-like video sequences.

Conclusion and feature direction

To mitigate the catastrophic impact of wildfires, it is critical to correctly and rapidly detect active flames in their early stages. There are a very few studies that focus on monitoring ongoing flames in near real-time using deep learning methods. In this work, we investigated the transfer learning of pre-trained models for detecting forest fire/smoke. We used the models to extract features and fine-tune them. The results indicate that the Xception-based model outperformed all other models with 98.72% accuracy. To preserve the characteristics of the old dataset, we employed LwF and found that it outperforms feature extraction. More interestingly, fine-tuning the new task with LwF performed comparatively well on original dataset when using fine-tuned parameters. Recent studies indicate that it is critical to detect fire mishaps quickly and accurately in their early stages to prevent them from spreading. As a consequence, we want to continue our study in this field and enhance our findings. In the future, we plan to apply the latest CNN models to rapidly identify fire occurrences with a low rate of false positives. Further, we like to explore more on LwF and multitask learning.

Authors' contributions

All authors contributed to the development of this manuscript. The authors read and approved the final manuscript.

Funding

This work was supported by a Korea Environmental Industry & Technology Institute (KEITI) grant funded by the Korean government (Ministry of the Environment), Project No. RE202101551, the development of IoT-based technology for collecting and managing big data on environmental hazards and health effects.

This research was also supported by an Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korean government (MIST) (No. 2019-0-00135, Implementation of 5G-based Smart Sensor Verification Platform) and partially supported by the Institute of Information and Communications Technology Planning and Evaluation (IITP) funded by the Korea Government, Ministry of Science and ICT(MSIT) (Building a Digital Open Lab as open innovation platform) under Grant 2021-0-00546.

Availability of data and materials

The datasets used and/or analyzed during the current study are available from the corresponding author on reasonable request.

Declarations

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Received: 7 November 2022 Accepted: 12 December 2022
Published online: 17 February 2023

References

Abid, F. 2021. A survey of machine learning algorithms based forest fires prediction and detection systems. *Fire Technology* 57 (2): 559–590.

Al-Duryi, M. H. A. 2022. *Design and analysis of forest fire detection system using image processing technique*. Altınbaş Üniversitesi/Lisansüstü Eğitim Enstitüsü.

Anh, N. D., P. Van Thanh, D. T. Lap, N. T. Khai, T. Van An, T. D. Tan, N. H. An, and D. N. Dinh. 2022. Efficient forest fire detection using rule-based multi-color space and correlation coefficient for application in unmanned aerial vehicles. *KSII Transactions on Internet and Information Systems (TIIS)* 16 (2): 381–404.

Arif, M., K. Alghamdi, S. Sahel, S. Alosaimi, M. Alsahaft, M. Alharthi, and M. Arif. 2021. Role of machine learning algorithms in forest fire management: a literature review. *J Robotics Autom* 5 (1): 212–226.

Bari, A., T. Saini, and A. Kumar: Fire detection using deep transfer learning on surveillance videos, in Editor (Ed.)^(Eds.): 'Book Fire detection using deep transfer learning on surveillance videos' (IEEE. 2021. edn.), pp. 1061–1067.

Best, N., J. Ott, and E. J. Linstead. 2020. Exploring the efficacy of transfer learning in mining image-based software artifacts. *Journal of Big Data* 7 (1): 1–10.

Bot, K., and J. G. Borges. 2022. A systematic review of applications of machine learning techniques for Wildfire Management decision support, Inventions, 7 (1): 15.

Bouguettaya, A., H. Zarzour, A. M. Taberkit, and A. Kechida. 2022. A review on early wildfire detection from unmanned aerial vehicles using deep learning-based computer vision algorithms. *Signal Processing* 190: 108309.

Castelli, M., L. Vanneschi, and A. Popovič. 2015. Predicting burned areas of forest fires: an artificial intelligence approach. *Fire ecology* 11 (1): 106–118.

Cheng, X. 2021. Research on application of the feature transfer method based on fast R-CNN in smoke image recognition, Advances in Multimedia. 2021.

Chollet, F. 2017. Xception: deep learning with depthwise separable convolutions, in Editor (Ed.)^(Eds.): 'Book Xception: Deep learning with depthwise separable convolutions'. edn.), 1251–1258.

Dampage, U., L. Bandaranayake, R. Wanasinghe, K. Kottahachchi, and B. Jayasanka. 2022. Forest fire detection system using wireless sensor networks and machine learning. *Scientific reports* 12 (1): 1–11.

de Venâncio, P. V. A., A. C. Lisboa, and A. V. Barbosa. 2022. An automatic fire detection system based on deep convolutional neural networks for low-power, resource-constrained devices, Neural Computing and Applications, 1–20.

Dogan, S., P. D. Barua, H. Kutlu, M. Baygin, H. Fujita, T. Tuncer, and U. R. Acharya. 2022. Automated accurate fire detection system using ensemble pretrained residual network. *Expert Systems with Applications* 203: 117407.

Fang, Q., Z. Peng, and P. Yan. 2022. Fire detection and localization method based on Deep Learning in Video Surveillance. In *)^(eds.): 'Book Fire detection and localization method based on Deep Learning in Video Surveillance'*, ed. Editor (, 012024. IOP Publishing. edn).

Fouda, M. M., S. Sakib, Z. M. Fadlullah, N. Nasser, and M. Guizani. 2022. *A lightweight hierarchical AI model for UAV-enabled edge computing with forest-fire detection use-case*. IEEE Network.

Friggens, M. M., R. A. Loehman, C. I. Constan, and R. R. Kneifel. 2021. Predicting wildfire impacts on the prehistoric archaeological record of the Jemez Mountains, New Mexico, USA, Fire Ecology. 17 (1): 1–19.

Guede-Fernández, F., L. Martins, R. V. de Almeida, H. Gamboa, and P. Vieira. 2021. A deep learning based object identification system for forest fire detection. *Fire* 4 (4): 75.

Ha, V. K., J. Ren, X. Xu, S. Zhao, G. Xie, and V. M. Vargas. 2018. Deep learning based single image super-resolution: a survey. In *)^(eds.): 'Book Deep learning based single image super-resolution: a survey'*, ed. Editor (, 106–119. Springer. edn).

Hu, Y., J. Zhan, G. Zhou, A. Chen, W. Cai, K. Guo, Y. Hu, and L. Li. 2022. Fast forest fire smoke detection using MVMNet. *Knowledge-Based Systems* 241: 108219.

Jeon, M., H.-S. Choi, J. Lee, and M. Kang. 2021. Multi-scale prediction for fire detection using convolutional neural network. *Fire Technology* 57 (5): 2533–2551.

Kaulage, A., S. Rane, and S. Dhore. 2022. *Satellite Imagery-Based wildfire detection using deep learning: 'Data Science'*. 213–220. Chapman and Hall/CRC.

Kizilkaya, B., E. Ever, H. Y. Yatbaz, and A. Yazici. 2022. An effective forest fire detection framework using heterogeneous wireless multimedia sensor networks, ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM). 18 (2): 1–21.

Ko, B. C., K.-H. Cheong, and J.-Y. Nam. 2009. Fire detection based on vision sensor and support vector machines. *Fire Safety Journal* 44 (3): 322–329.

Kong, S. G., D. Jin, S. Li, and H. Kim. 2016. Fast fire flame detection in surveillance video using logistic regression and temporal smoothing. *Fire Safety Journal* 79: 37–43.

Krizhevsky, A., I. Sutskever, and G. E. Hinton. 2017. Imagenet classification with deep convolutional neural networks. *Communications of the ACM* 60 (6): 84–90.

LeCun, Y., L. Bottou, Y. Bengio, and P. Haffner. 1998. Gradient-based learning applied to document recognition, Proceedings of the IEEE. 86 (11): 2278–2324.

Li, P., and W. Zhao. 2020. Image fire detection algorithms based on convolutional neural networks. *Case Studies in Thermal Engineering* 19: 100625.

Li, S., Q. Yan, and P. Liu. 2020. An efficient fire detection method based on multiscale feature extraction, implicit deep supervision and channel attention mechanism. *IEEE Transactions on Image Processing* 29: 8467–8475.

Luo, Y., L. Zhao, P. Liu, and D. Huang. 2018. Fire smoke detection algorithm based on motion characteristic and convolutional neural networks. *Multimedia Tools and Applications* 77 (12): 15075–15092.

Mahmoud, H. A. H., A. H. Alharbi, and N. S. Alghamdi. 2022. Time-efficient fire detection convolutional neural network coupled with transfer learning, INTELLIGENT AUTOMATION AND SOFT COMPUTING. 31 (3): 1393–1403.

Majid, S., F. Alenezi, S. Masood, M. Ahmad, E. S. Gündüz, and K. Polat. 2022. Attention based CNN model for fire detection and localization in real-world images. *Expert Systems with Applications* 189: 116114.

Mao, W., W. Wang, Z. Dou, and Y. Li. 2018. Fire recognition based on multichannel convolutional neural network. *Fire technology* 54 (2): 531–554.

Muhammad, K., J. Ahmad, I. Mehmood, S. Rho, and S. W. Baik. 2018. Convolutional neural networks based fire detection in surveillance videos. *Ieee Access : Practical Innovations, Open Solutions* 6: 18174–18183.

Muhammad, K., J. Ahmad, and S. W. Baik. 2018. Early fire detection using convolutional neural networks during surveillance for effective disaster management. *Neurocomputing* 288: 30–42.

Namozov, A., and Y. Im Cho. 2018. An efficient deep learning algorithm for fire and smoke detection with limited data. *Advances in Electrical and Computer Engineering* 18 (4): 121–128.

Nguyen, H. V., T. X. Pham, and C. N. Le. 2021. Real-time long short-term glance-based fire detection using a CNN-LSTM neural network. *International Journal of Intelligent Information and Database Systems* 14 (4): 349–364.

Park, M., and B. C. Ko. 2020. Two-step real-time night-time fire detection in an urban environment using static ELASTIC-YOLOv3 and temporal fire-tube. *Sensors (Basel, Switzerland)* 20 (8): 2202.

Qin, Y.-Y., J.-T. Cao, and X.-F. Ji. 2021. Fire detection method based on depthwise separable convolution and yolov3. *International Journal of Automation and Computing* 18 (2): 300–310.

Saeed, F., A. Paul, P. Karthigaikumar, and A. Nayyar. 2020. Convolutional neural network based early fire detection. *Multimedia Tools and Applications* 79 (13): 9083–9099.

Saeed, F., A. Paul, A. Rehman, W. H. Hong, and H. Seo. 2018. IoT-based intelligent modeling of smart home environment for fire prevention and safety. *Journal of Sensor and Actuator Networks* 7 (1): 11.

Seydi, S. T., V. Saeidi, B. Kalantar, N. Ueda, and A. A. Halin. 2022. Fire-Net: a deep learning framework for active forest fire detection, Journal of Sensors. 2022.

Schmidhuber, J. 2015. Deep learning in neural networks: an overview. *Neural networks* 61: 85–117.

Sharma, J., O.-C. Granmo, M. Goodwin, and J. T. Fidje. 2017. Deep convolutional neural networks for fire detection in images. In *)^(eds.): 'Book Deep convolutional neural networks for fire detection in images'*, ed. Editor (, 183–193. Springer. edn).

Simonyan, K. and Zisserman. 2015. A Very deep convolutional networks for large-scale image recognition. Proceedings of the International Conference on Learning Representations

Sousa, M. J., A. Moutinho, and M. Almeida. 2020. Wildfire detection using transfer learning on augmented datasets. *Expert Systems with Applications* 142: 112975.

Szegedy, C., W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. 2015. Going deeper with convolutions, in Editor (Ed.)^(Eds.): 'Book Going deeper with convolutions'. edn.), pp. 1–9.

Tian, L., C. Fan, Y. Ming, and Y. Jin: Stacked PCA network (SPCANet): an effective deep learning for face recognition, in Editor (Ed.)^(Eds.): 'Book Stacked

PCA network (SPCANet): an effective deep learning for face recognition' (IEEE. 2015. edn.), pp. 1039–1043.

Yang, H., H. Jang, T. Kim, and B. Lee. 2019. Non-temporal lightweight fire detection network for intelligent surveillance systems. *Ieee Access : Practical Innovations, Open Solutions* 7: 169257–169266.

Yang, S., S. Zhang, X. Chen, J. Li, E. Li, and W. Chen: A fire detection method based on computer vision, in Editor (Ed.)^(Eds.): 'Book A Fire Detection Method based on Computer Vision' (IEEE. 2022. edn.), pp. 11–15.

Zhao, L., L. Zhi, C. Zhao, and W. Zheng. 2022. Fire-YOLO: a small target object detection method for fire inspection. *Sustainability* 14 (9): 4930.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.