Journal of Computational Design and Engineering, 2022, 9(2), 793–804

https://doi.org/10.1093/jcde/qwac027
Journal homepage: www.jcde.org

RESEARCH ARTICLE

A video-based SlowFastMTB model for detection of small amounts of smoke from incipient forest fires

Minseok Choi, Chungeon Kim and Hyunseok Oh *

School of Mechanical Engineering, Gwangju Institute of Science and Technology, Gwangju 61005, South Korea

*Corresponding author. E-mail: hsoh@gist.ac.kr https://orcid.org/0000-0002-6127-561X

Abstract

This paper proposes a video-based SlowFast model that combines the SlowFast deep learning model with a new boundary box annotation algorithm. The new algorithm, namely the MTB (i.e., the ratio of the number of Moving object pixels To the number of Bounding box pixels) algorithm, is devised to automatically annotate the bounding box that includes the smoke with fuzzy boundaries. The model parameters of the MTB algorithm are examined by multifactor analysis of variance. To demonstrate the validity of the proposed approach, a case study is provided that examines real video clips of incipient forest fires with small amounts of smoke. The performance of the proposed approach is compared with those of existing deep learning models, including convolutional neural network (CNN), faster region-based CNN (faster R-CNN), and SlowFast. It is demonstrated that the proposed approach achieves enhanced detection accuracy, while reducing false negative rates.

Keywords: forest fire; smoke detection; deep learning; early detection; annotation

1. Introduction

Recently, forest fires have occurred more frequently; the increase is the result of multiple factors, including anthropogenic climate change (Williams *et al.*, 2019). Forest fires have serious ecological, societal, and economic impacts, including human respiratory diseases and deforestation (Siscawati, 1998). Severe forest fires can even damage forest ecosystem resilience (Coop *et al.*, 2020). If incipient forest fires can be detected, the potential damage and economic loss of the fires can be reduced significantly. Therefore, early detection of forest fires is of great research and societal importance (Alkhatib, 2014).

Without relying on the naked eye, which is vulnerable to human error and fatigue, forest fires can be monitored by using infrared, near-infrared, LiDAR, and CCD/CMOS sensors (Alkhatib, 2014). For example, infrared and near-infrared sensors have been shown to be effective, since they are sensitive to small (less than 0.1 m^2) and cool (less than 600 K) incipient forest fires (Thomas & Nixon, 1993). However, infrared, near-infrared, and LiDAR sensors have relatively short measuring distances and are expensive (Starr & Lattimer, 2012). To overcome these shortcomings, numerous studies have reported the development of forest fire detection systems using CCD/CMOS sensors (i.e., using visible images), which are inexpensive and versatile (Han & Lee, 2006).

Forest fires are characterized by flame and smoke. When a forest fire occurs, smoke precedes the flame. Therefore, smoke can be an early indicator for detecting incipient forest fires (Li *et al.*, 2013). Nevertheless, the detection of incipient forest fires based on visible images is challenging due to the temporal evolution of smoke. Likewise, when examining flames, the appearance of flames can vary tremendously in terms of colors, textures, shapes, and occlusions (Yuan *et al.*, 2015). A decade ago, automated forest fire smoke detection systems began to be developed (Xu & Xu, 2007). Handcrafted features of these systems were designed by human experts (Lee *et al.*, 2009; Shukla & Pal, 2009). Then, machine learning methods, such as Gaussian mixture model (GMM), support vector machine, histogram of oriented gradients, and histogram of optical flow, were incorporated to determine the occurrence of incipient

Received: 30 October 2021; **Revised:** 12 February 2022; **Accepted:** 4 March 2022

forest fires (Xiong *et al.*, 2007; Chen *et al.*, 2013; Park *et al.*, 2013). These machine-learning-based approaches are labor-intensive and costly in their extraction of handcrafted features (Koo & Shin, 2018). Their performance depends on the subjective combination of handcrafted features (Oh *et al.*, 2018). Deep learning, which has shown excellent performance in various research fields, such as speech recognition (Devlin *et al.*, 2019), natural language processing (Vaswani *et al.*, 2017), and image detection (Tan & Le, 2019), has the potential to overcome the shortcomings of the machine-learning-based approaches. Recently, existing deep learning models are evolving further to detect variants of malicious code (Cui *et al.*, 2018), to find the optimal architecture of probabilistic neural network and extreme learning machines (Wang *et al.*, 2016; Yi *et al.*, 2016), and to improve the performance of wavelet neural networks in target treat assessment (Wang *et al.*, 2013), etc. For forest fire detection, 2D-data-based deep learning algorithms using forest fire visible images were studied based on a convolutional neural network (CNN; Sun *et al.*, 2021). The CNN-based algorithm proposed by Sun *et al.* was designed to extract complex features in a supervised manner from a large number of visible images of smoke data.

To detect the initiation of forest fire by capturing the difference between two images, change detection algorithms were proposed. Existing approaches for change detection can be grouped into pixel-based and object-based. Liu *et al.* (Liu *et al.*, 2021) provided an excellent review of pixel-based and object-based approaches for change detection. Pixel-based detection approaches extract features from pixels and assign the different classification label to the pixels. For example, small-scale forest fires could be detected using a deep learning model trained using RGB images (Tang *et al.*, 2020). These studies employed high-resolution RGB images (e.g., 4K). In reality, it can be difficult to obtain high-resolution images due to practical reasons. Ghali *et al.* (Ghali *et al.*, 2021) worked to detect wildfire using deep vision transformers with low-resolution RGB images.

Different from the pixel-based detection approaches, object-based detection approaches regard a group of homogeneous pixels as a unit. The unit can be generated by unsupervised algorithms for image segmentation. As the object-based detection approaches treat a single object as a unit for a given task, all the pixels within the object are assigned with the same classification label. This leads to a considerable reduction of the "salt-and-pepper" noise that is commonly observed from the results of pixel-based detection approaches (Chen *et al.*, 2012). Owing to the potential advantages of the object-based detection approaches over the pixel-based detection approaches, object-based methods have been increasingly used for detecting land cover changes in natural environments (Wang *et al.*, 2018), assessing disaster damages (Gong *et al.*, 2012), and examining forest disturbance (Healey *et al.*, 2018). The success of object-based methods in detecting land cover changes led to its extensive adoption in change detection applications. For example, three-dimensional (3D) CNN and YOLOv3 were developed to detect forest fire and smoke in recent years (Jiao *et al.*, 2019; Kim & Lee, 2019; Lin *et al.*, 2019). These studies presented its superior performance over existing methods, although forest fire detection using deep learning techniques is still in its infancy.

The object detection algorithms developed required annotations for locating the position of the smoke. To reduce the effort required to manually label fuzzy smoke boundaries, synthetic images were generated to train the algorithms by overlapping real and synthetic smoke images with forest backgrounds (Zhang *et al.*, 2018; Yuan *et al.*, 2019). Synthetic images can be useful to generate big datasets that emulate forest fires subject to various environmental conditions (Goncalves *et al.*, 2020). However, it is apparent that synthetic smoke images are different from real smoke images in terms of the motion, density, and nature of the background forest. For example, a deep CNN trained using synthetic smoke images showed a significant performance drop, mainly attributed to the discrepancy between synthetic and real smoke images (Xu *et al.*, 2019). In summary, previous studies have not presented a robust approach to annotate a large amount of real smoke images for training object detection algorithms.

The ultimate goal of detecting smoke accurately and precisely is to avoid and mitigate damage from forest fires. As presented earlier, deep-learning-based smoke detection methods proposed previously have been trained and tested using images in which most of the pixels in the images were occupied by smoke. This is not the case for real camera images of incipient forest fires. If a sufficient amount of smoke pixels is found from the real camera images, it indicates that forest fires have already led to significant damage. In general, incipient forest fire smoke is difficult to detect since it is represented by a small number of pixels in the overall camera images. While existing research studies have sought approaches to detect smoke in forest background images, there is little research to date on the detection of small amounts of smoke in these images.

To fill this research gap, this paper proposes a video-sequence-based methodology that combines a novel algorithm for positioning the annotation box with a SlowFast model for detecting smoke from temporal video sequences. A novel algorithm, called MTB (i.e., the ratio of the number of Moving object pixels To the number of Bounding box pixels), is designed to automatically annotate fuzzy smoke boundaries from forest background images. The hyperparameters of the MTB algorithm are selected using the analysis of variance (ANOVA). The SlowFast model was originally developed to detect human action. In this research, the MTB algorithm is substituted for the human detection module in the SlowFast model. To evaluate the performance of the proposed method, various existing methods are compared. We also attempt to interpret the validity of the results using the deep-learning visualization technique called gradient-weighted class activation maps (Grad-CAM).

The contributions of this paper can be summarized in two primary ways. First, the new MTB algorithm is proposed to automatically annotate smoke with fuzzy boundaries. The MTB algorithm detects the moving object (i.e., smoke) in RGB video images. Second, a new method, namely SlowFastMTB, is proposed to detect the smoke from incipient forest fires. The original SlowFast model developed for action recognition is modified to detect small amounts of smoke by incorporating the MTB algorithm.

The remainder of the paper is organized as follows: Section 2 provides an overview of the theoretical background of the methods used in this study. Section 3 proposes the video-sequence methodology that combines the MTB algorithm with the SlowFast model for early forest fire smoke detection. In Section 4, the experimental setup is presented. Section 5 shows the performance of the proposed methodology. Finally, Section 6 offers the conclusions of the paper and discusses future work.

2. Theoretical Background

This section describes the theoretical background of the proposed approach. In this study, residual networks (ResNet) are used to extract relevant features from visible images. Compared

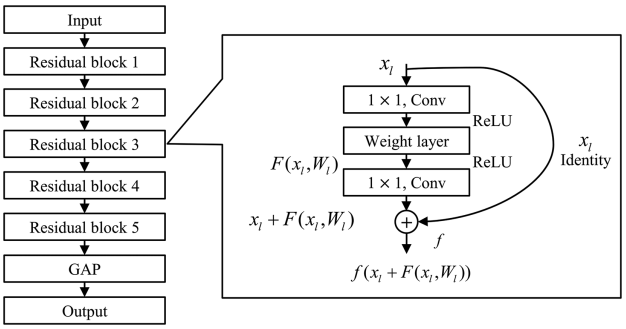


Figure 1: Residual network architecture.

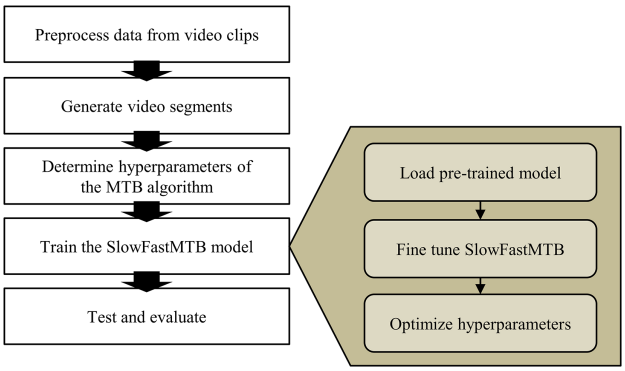


Figure 2: Overall procedure for forest fire smoke detection.

to previous networks, including AlexNet, VGG, and GoogleNet, ResNet was proposed to address the gradient vanishing and exploding problems (He *et al.*, 2016a). Gradient vanishing is a phenomenon in which the gradient gets smaller as it reaches the input during back propagation. The deeper the neural network, the more the gradient vanishes. In contrast, in the case of a gradient explosion, the gradient grows abnormally as it reaches the input layer. Gradient vanishing and exploding are reasons why learning effectiveness is reduced in a neural network. The ResNet architecture innovatively includes the residual block and the bottleneck architecture to overcome the problems.

ResNet realizes residual mapping by adding a skip connection to the existing layer, as shown in Fig. 1. A module with a skip connection added is called a residual block. The feature output of a single residual block (y_l) can be expressed as

$$\begin{aligned} y_l &= x_l + F(x_l, W_l) \\ x_{l+1} &= f(y_l) \end{aligned} \tag{1}$$

where x_l is the feature output of the l^{th} residual unit; W_l is the weights of the l-th residual unit; F is the residual function; and f is the activation function of the rectified linear unit. If f is assumed to be identity, the output of the residual block can be expressed as (He *et al.*, 2016b)

$$x_L = x_l + \sum_{i=l}^{L-1} F(x_i, W_i) \tag{2}$$

where x_L is the output of the L^{th}neural unit. x_L is expressed as the summation of the output of the shallow layer (i.e., x_l) and the output of the residual function between units L and l. The summation notation in equation (2) prevents gradient vanishing from occurring during the backward propagation. To understand this, the gradient of the loss function ε is calculated through the chain rule.

$$\begin{aligned} &\frac{\partial \varepsilon}{\partial x_l} = \frac{\partial \varepsilon}{\partial x_L}\frac{\partial x_L}{\partial x_l} = \frac{\partial \varepsilon}{\partial x_L}(G_1 + G_2) \\ &\text{where} \quad G_1 = 1, \quad G_2 = \frac{\partial}{\partial x_l}\sum_{i=l}^{L-1} F(x_i, W_i) \end{aligned} \tag{3}$$

The gradient of the loss function is composed of two terms, G_1 and G_2. The G_1 term guarantees that the gradient directly propagates, regardless of the neural layer. For a mini-batch sample, if the G_2 term is minus one, the gradient can be vanishing (i.e., the gradient is zero). However, in general, it is unlikely that all mini-batch samples will be minus one. In other words, the possibility of gradient vanishing is low (He *et al.*, 2016b).

Deep neural networks go deeper; thus, additional calculation is required to train the networks. ResNet uses a bottleneck architecture that significantly reduces the number of operations while maintaining performance. The bottleneck architecture was proposed in Inception v1 (Szegedy *et al.*, 2015). A one-by-one convolution is put into both ends of the existing residual block as shown in Fig. 1. The depth of the feature is reduced and then increased again through the one-by-one convolution operation. This reduces the computational costs significantly as ResNet goes deeper.

3. Proposed Model

This section presents a new model that is capable of detecting small-scale early forest fire smoke. The proposed model combines the new MTB algorithm with the SlowFast network. Section 3.1 provides an overview of the procedure for developing the proposed model. Section 3.2 describes the MTB algorithm, which automatically annotates the smoke. Section 3.3 explains how the MTB algorithm is integrated with the SlowFast network.

3.1 Overall procedure

The overall procedure of this study is shown in Fig. 2. First, the dataset is preprocessed to enhance the quality of images from video clips. For example, a Gaussian blur filter can be imposed on the video clip images to reduce detail and noise. Second, the preprocessed video clip images are divided into segments of specific time increments. Segments of specific time increments are used for training and evaluation of deep learning models with k-fold cross-validation. Third, the hyperparameters of the MTB algorithm are selected through ANOVA. At this step, metaheuristic algorithms such as genetic algorithms can be incorporated. Fourth, the SlowFastMTB model is trained. During training, the weights of the pretrained model are loaded, the classifier for the smoke is initialized, the SlowFastMTB model is fine-tuned, and the hyperparameters of the SlowFastMTB model are optimized. Finally, the trained SlowFastMTB model is tested and analysed via visualization.

3.2 New bounding box detection algorithm: MTB

Some object detection targets (e.g., humans) have clear boundaries. In these cases, the ground truth can be labeled based on the boundaries. For example, see GMM (Burić *et al.*, 2018). However, smoke has no clear boundaries. It is not possible to label the bounding box of smoke with existing object detection criteria. To address this problem, this study proposes the MTB algorithm.

The MTB algorithm finds moving objects through pixels that exceed the threshold in the difference between frames, whose concept is similar to that of GMM. However, GMM has limitations when detecting moving objects with fuzzy boundaries. A particular area of the smoke is dense enough to cover the background or less dense and semitransparent. Due to the characteristics, the GMM method can be problematic when detecting a specific area of smoke rather than finding the entire smoke, and is sensitive to threshold.

The MTB algorithm relaxes the bounding box that includes the entire smoke even when a specific part of a fuzzy moving object is detected. The key idea of the MTB algorithm is to adjust the size of the detected area (i.e., bounding box) using the index, called r_{MTB}:

$$r_{MTB} = \frac{N_M}{N_B} \tag{4}$$

where N_M is the number of moving object pixels in a particular bounding box and N_B is the total number of pixels in the identical bounding box. r_{MTB} for the bounding box detected is expressed as $r_{MTB,dt}$, whereas r_{MTB} for the bounding box desired is described as $r_{MTB,ds}$. The bounding box detected is resized to that with a desired target value. Then, within the resized bounding box, an object detection task is conducted with a deep learning model. With the development of the r_{MTB} index, it became feasible to detect the entire area of a moving object with fuzzy boundaries even if a hard threshold is used.

The pseudocode for the detailed procedure is specified in Algorithm 1. The MTB algorithm outputs smoke's resized bounding box (B_r) when video sequences (V) are the inputs. To initiate the MTB algorithm, three hyperparameters including time interval (Δt), threshold (T), and desired MTB value ($r_{MTB,ds}$) should be determined by the user. Once inputs are prepared with a set of hyperparameters, the MTB algorithm can be initiated to locate bounding boxes for individual visual images from video clips. The algorithm is comprised of four detailed steps: (i) pixel-wise subtraction between consecutive frames, (ii) threshold-based classification, (iii) bounding box acquisition, and (iv) bounding box scaling. First, the pixel-wise subtraction between two consecutive frames is conducted using $V(t)$ and $V(t + \Delta t)$. When the time is t seconds, the video frame is recorded as $V(t)$. When the time elapsed by Δt seconds, the video frame is recorded as $V(t + \Delta t)$. Second, when the pixel-wise difference between the two frames exceeds the threshold, the color of the pixel is changed to white. Otherwise, it is changed to black. White-colored pixels indicate a moving object, whereas black-colored pixels correspond to the background. Third, the smallest bounding box containing all of the white-colored pixels can be obtained. Fourth, the size of the bonding box is scaled. The $r_{MTB,dt}$ is calculated by dividing the number of white pixels in the smallest bounding box (B) by the number of total pixels in the identical box. It is resized through $r_{MTB,ds}$ and $r_{MTB,dt}$ so that B has $r_{MTB,ds}$. The final output is obtained: resized bounding box of smoke (B_r).

An example of the MTB algorithm is shown in Fig. 3. The MTB algorithm starts with a frame when the time is 10 seconds. When the time interval is 5 seconds, a frame of 15 seconds is selected. The absolute value of the difference between the elements of two frames is calculated. When the difference exceeds the threshold of 19, the value is changed to 255; otherwise, it is changed to zero. The value of 255 is shown in white and the value of zero is shown in black. The value of 255 is considered to be the moving object and the value of zero is considered to be the background. In this example, $r_{MTB,dt}$ is 0.5 and $r_{MTB,ds}$ is 0.05. Therefore, B_r is obtained by magnification 10 times that of B.

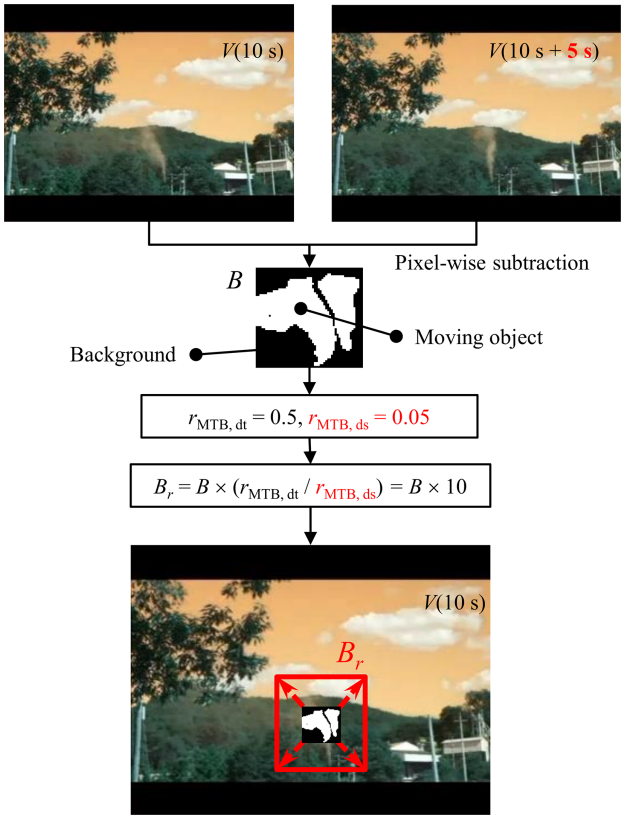


Figure 3: An example for implementing the MTB algorithm.

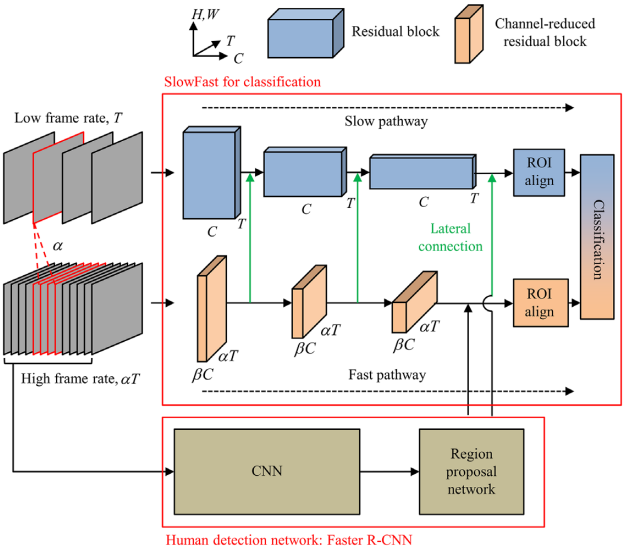


Figure 4: Original SlowFast model for human action detection.

3.3 SlowFastMTB

The SlowFast network is a two-way, end-to-end network for human action recognition (Feichtenhofer *et al.*, 2019). As illustrated in Fig. 4, it consists of two main parts: (i) slow and fast pathways with lateral connections and (ii) a human detection network.

The slow and fast pathways are the main component to recognize human action. The slow pathway is a 3D CNN model (e.g., ResNet) with a spatiotemporal volume. The slow pathway

Algorithm 1: MTB algorithm

Input: Smoke video frames (V)
Output: Resized bounding box of smoke (B_r)
Require: Time interval (Δt), threshold (T), desired MTB value ($r_{MTB,ds}$)

Initialize the time-step: $t \leftarrow 0$
Initialize the pixel-wise difference between frames: $e \leftarrow 0$
Initialize the detected bounding box MTB: $r_{MTB,dt} \leftarrow 0$

while all video frames are performed **do**
  // Step 1. pixel-wise subtraction between consecutive frames
  Pixel-wise subtraction of the video frame at t from that at $t + \Delta t$: $e \leftarrow V(t + \Delta t) - V(t)$
  // Step 2. threshold-based classification
  If at least one pixel from e is greater than T:
    Change the pixel whose e is greater than T into a white pixel
  Else:
    Change the pixel whose e is equal to or smaller than T into a black pixel
  // Step 3. bounding box acquisition
  Assign the smallest box that contains all the white pixels: B
  // Step 4. bounding box scaling
  Compute the ratio: $r_{MTB,dt} \leftarrow$ number of white pixels/number of smallest box pixels
  Resize the bounding box of $V(t)$: $B_r \leftarrow B \times r_{MTB,dt}/r_{MTB,ds}$
  $t \leftarrow t + 1$
End
return B_r

captures spatial semantics using low-speed frames. The fast pathway is a 3D CNN model (e.g., ResNet) with the same structure as the slow pathway but with a reduced number of channels. The fast pathway captures human action using high-speed frames. Both the slow and fast pathways use two hyperparameters. First, the temporal ratio between slow and fast pathway (α) is used to downsample the input data. Second, the channel scaling ratio (β) is to control the number of channels in the deep learning model. The frame rate ratio of the fast and slow pathways is determined with the temporal ratio. The amount of computation of the fast pathway is reduced through the use of a channel scaling ratio for low channel capacity. The slow and fast pathways extract different spatiotemporal information. The lateral connection is a one-way connection that merges information from the slow pathway and the fast pathway. The features of the fast pathway are connected directionally to those of the slow pathway. Since the outputs of the fast pathway and the slow pathway are different, it is necessary to match the shape of the feature to merge information.

A human detection network is for the detection of an object. In the task of detecting a person in a video and classifying the person's action, an algorithm is required to detect the person. SlowFast uses faster R-CNN's (faster region-based CNN) region proposal network (RPN) to detect people (Ren *et al.*, 2015). The RPN searches the entire image based on k anchor boxes in the extracted feature map. Next, the RPN predicts the object's bounding box and class. After sorting the results into the order of highest classification probability, the nonmaximum suppression is used to anchor boxes so that the region of interest is obtained.

The architecture of the proposed SlowFastMTB model is shown in Fig. 5. The original SlowFast network incorporated faster R-CNN to locate a person. Then, the image captured by the bounding box is used to classify various human actions. Thus, SlowFast detects people and categorizes their actions. For the detection of smoke from the background images, smoke does not require the classification of the movement. Only the detection of smoke is necessary. It is not appropriate to implement the original SlowFast network for smoke detection. To this end, the faster R-CNN of the original SlowFast network was replaced with the newly proposed MTB algorithm.

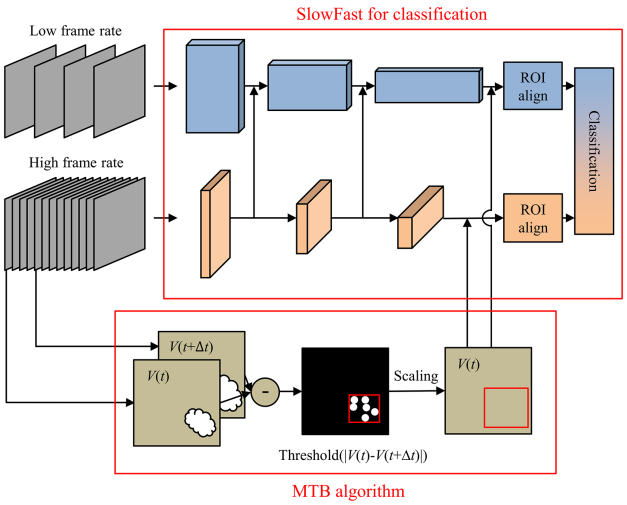


Figure 5: Proposed SlowFastMTB model for the early detection of forest fire smoke.

4. Experimental Setup

This section presents the experimental setup. Section 4.1 shows four datasets that were used to determine the hyperparameters of the proposed MTB algorithms and evaluate the performance of the proposed SlowFastMTB model. Section 4.2 describes how

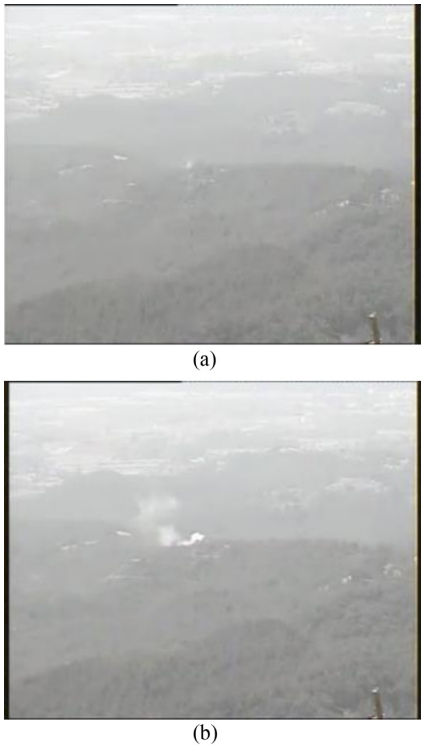


Figure 6: Representative visual image from Video A: (a) incipient smoke and (b) the smoke occupies 1.02% out of the image.

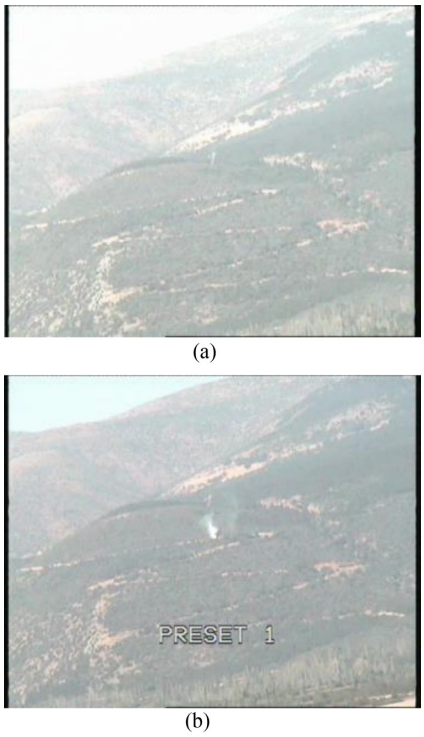


Figure 7: Representative visual image from Video B: (a) nonexistence of smoke and (b) the smoke occupies 0.48%.

Figure 8: Representative visual image from Video C; the smoke occupies 15.74%.

to determine the optimal values of the hyperparameters. Section 4.3 depicts the implementation details of the proposed model for the detection of incipient forest fires.

4.1 Dataset

Visual images of evolving incipient forest fires were captured from a dataset of Video A (46). As shown in Fig. 6, the smoke in the visual images evolves from nonsmoke to smoke that occupies 1.02% out of the image, while the position of the fire does not change. A Gaussian blur filter is used to remove noise from the video. The number of frames that contained smoke and the number that did not contain smoke were 1000 and 1000, respectively. From the dataset of Video A, the number of video segments was 2000 in total (i.e., 1000 with smoke and 1000 without smoke). The size of the original images was 352 by 288 pixels (width by height). In this study, the original images were resized to 224 by 224 pixels. This dataset was used to determine hyperparameters of the proposed MTB algorithm.

The second dataset, from Video B (46), is more complicated than that from Video A. As shown in Fig. 7, Video B starts with no smoke. Then, smoke evolves to smoke that occupies 0.48% out of the image. The background images from Video B include smoke-like objects, such as earth, rocks, and roads. It should be noted that all images from Video B have an identical background, whether there is smoke or not. The size of the original images was 720 by 516 pixels (width by height). In this study, the original images were resized to 224 by 224 pixels. To avoid a data imbalance problem, a balanced amount of data was used. The number of frames that contained smoke and the number that did not contain smoke were 4050 and 4050, respectively. When the frame rate was assumed to be 25 frames per second, the 4050 frames correspond to 162 seconds with one second window. Then, video clips of one second in length were regarded as a single input data for training the video-based deep learning models, including SlowFast and SlowFastMTB. These one-second-long video clips (i.e., 4050 video segments) were randomly divided into five datasets for fivefold cross-validation. In summary, from the dataset of Video B, the number of video segments was 8100 in total (i.e., 4050 with smoke and 4050 without smoke).

Additional datasets of wildfire video clips (i.e., videos C and D) (47) were used to evaluate the performance of the trained SlowFastMTB model. As presented in Fig. 8, smoke is located on the background of the mountain, where the smoke occupies 15.74% of the image. The original video clips with the size of 320 by 240 pixels (width by height) were resized to 224 by 224 pixels. To increase the amount of data, the video segments of 200 were generated using one second window. The Video D dataset is different from the Video C dataset in that smoke is located on the background of moving clouds. As shown in Fig. 9, the smoke occupies 1.92% of the image.

Figure 9: Representative visual image from Video D; the smoke occupies 1.92%.

Table 1: Setting for multifactor ANOVA.

Factor (model parameter of the MTB algorithm)	Level		
Time interval (seconds)	5	10	15
Threshold	17	19	21
$r_{MTB,ds}$	0.01	0.05	0.1

4.2 Parametric study of the MTB algorithm

The hyperparameters of the proposed SlowFastMTB model should be selected by the user. In this study, the hyperparameters can be divided into two groups: SlowFast model-related and MTB-related. The SlowFast model-related hyperparameters include batch size, dropout rate, momentum, weight decay, and learning rate. The MTB-related hyperparameters are time interval, threshold, and $r_{MTB,ds}$. An optimal set of the hyperparameters can be searched for using either swarm-based algorithms or evolutionary algorithms. There are numerous swarm-based algorithms and their variants inspired by the swarm behavior of honey bees, bats, and chickens (Feng *et al.*, 2021; Li *et al.*, 2021). Representative examples of the evolutionary algorithms are genetic algorithms, evolutionary programming, and differential evolution (Eiben & Smith, 2015; Gao *et al.*, 2020). Simple metaheuristic algorithms offer the best performance for large-scale problems, whereas a particular metaheuristic algorithm with an optimal set of model parameters can provide better performance for mid-scale or small-scale problems (Kim *et al.*, 2021). A tradeoff between the accuracy and the computational cost must be considered for the optimization problem to be solved. A comprehensive analysis of the exact hyperparameters is out of the scope of this paper. Thus, in this study, one of simple statistical tools, ANOVA, was employed.

A parametric study was performed to understand the effect of model parameters of the MTB algorithm on the accuracy of object detection. A three-factor, three-level ANOVA was conducted, as shown in Table 1. The default values were set to be the time interval of 10 seconds, the threshold of 19, and the $r_{MTB,ds}$ value of 0.05. The multifactor ANOVA was also designed to examine the interaction between the model parameters of the MTB algorithms. Faster R-CNN models with a VGG16 backbone were used as an object detection model. The faster R-CNN model was trained with the given combinations of model parameters. The detection threshold was set to be 0.8.

Using the trained faster R-CNN models, the test accuracy of Video A was evaluated. In ANOVA, the null hypothesis was that the model parameter does not have significant impact on the object detection accuracy. The boundary of the *P*-value for rejecting the null hypothesis was 0.01. As shown in Table 2, only the *P*-value of $r_{MTB,ds}$ was less than 0.01. It was indicated that the effect of $r_{MTB,ds}$ on the accuracy of the deep learning model cannot be ignored. Therefore, a value of 0.01 $r_{MTB,ds}$ with the highest accuracy can be adopted. Interaction effects between model parameters were analysed. As depicted in Table 3, only the *P*-value of the interaction between the time interval and $r_{MTB,ds}$ was less than 0.01. With the observations, as presented in Fig. 10, $r_{MTB,ds}$ and time interval values that correspond to the location with the highest accuracy were 0.01 and 5, respectively. From the parametric study using ANOVA, a set of model parameter values were determined, specifically the time interval of 5 seconds, the threshold of 19, and the $r_{MTB,ds}$ value of 0.01.

Table 2: ANOVA results of model parameters of the MTB algorithm.

	Levels	Average accuracy (%)	*P*-value
Time interval (seconds)	5	89.4	0.524
	10	91.4	
	15	92.4	
Threshold	17	92.2	0.656
	19	91.2	
	21	89.8	
$r_{MTB,ds}$	0.01	94.9	0.000
	0.05	91.5	
	0.1	86.8	

Table 3: ANOVA results for interaction between model parameters.

Interaction	*P*-value
Time interval × $r_{MTB,ds}$	0.003
Time interval × threshold	0.583
Threshold × $r_{MTB,ds}$	0.240

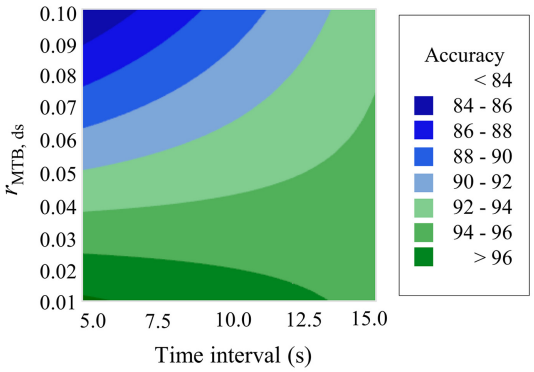


Figure 10: Interaction between the time interval and $r_{MTB,ds}$.

4.3 Implementation details

As discussed in Section 3, a pretrained ResNet50 was used as the backbone of the deep learning models. Image-based deep learning models, such as CNN and faster R-CNN, employed 2D ResNet50 pretrained using the ImageNet image datasets. The implementation of the pretrained ResNet50 model to the deep learning models was conducted in three steps. The first step was to randomly initialize the weights and biases of the fully

Table 4: Hyperparameter setting for fine-tuning the pretrained ResNet50 model.

Model	Hyperparameter	Value
ResNet50	Batch size	200
	Dropout rate	0.5
	Optimizer	Stochastic gradient descent
	Momentum	0.9
	Weight decay	10^{-7}
	Learning rate	0.01
	Loss function	Binary cross-entropy

Table 5: Hyperparameter setting for training the SlowFast model.

Model	Hyperparameter	Value
SlowFast	Temporal ratio (α)	4
	Channel scaling ratio (β)	0.125

connected layers of the pretrained model. The second step was to train the weights and biases by freezing the convolutional layers of the pretrained model. The final step was to conduct fine-tuning of all of the layers. The values of the hyperparameters used for fine-tuning the ResNet50 models are specified in Table 4.

Video-based deep learning models incorporated 3D ResNet50 pretrained using the AVA v2.2 video datasets. The pretrained 3D ResNet models were implemented with the identical three steps. The values of the hyperparameters of the SlowFast models are described in Table 5.

The deep learning models were coded using Python 3.8.8 and Pytorch 1.4.0 on the operating system of Linux 16.04 LTS. A desktop computer was used with an Intel Core i7-9800X (3.80 GHz) processor, 128 gigabytes of RAM (DDR4), and an NVIDIA GeForce RTX 2080 Ti graphics card. The training of the deep learning models was conducted with an NVIDIA graphics card, while the training results were visualized with an Intel processor.

5. Results and Discussion

The proposed MTB algorithm was designed to determine bounding boxes of smoke in frames from video clips. As presented in Fig. 11, it was observed that the bounding boxes of the smoke were located correctly. The proposed model (i.e., SlowFastMTB) was used to determine the occurrence of forest fires from the video clips by calculating the probability of the presence of smoke. For the situation shown in Fig. 11a, smoke was detected; the probability of smoke was determined to be 0.99 and the probability of there not being smoke was 0.01. Based on the analysis of the video clip, it was determined that a forest fire had occurred, since the probability of there being smoke in the image (i.e., 0.99) was higher than the detection threshold of 0.8. When the shape of the smoke evolved over time, the forest fire was detected in a robust manner. However, for the scenario shown in Fig. 11b, it was determined that a forest fire did not occur, since the probability of there being smoke in the image of 0.72 was lower than the detection threshold of 0.8. If the threshold is adjusted, smoke would be detected. Thus, the value of the threshold must be determined carefully. A low threshold value will increase the false positive (FP) rate, while decreasing the false negative (FN) rate, and vice versa.

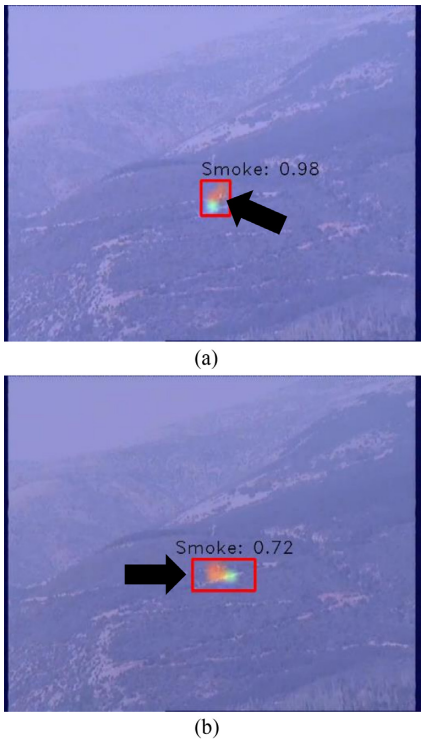


Figure 11: Representative results of the proposed SlowFastMTB model using the Video B dataset; (a) the small amount of smoke was detected correctly and (b) it was determined that the smoke was not detected since the probability of the smoke of "0.72" was less than the threshold of 0.8. Nonetheless, the Grad-CAM result presented that the location of actual smoke is correctly detected. It should be noted that the location of the actual smoke is indicated by the black arrow; the red box implies that the bounding box is predicted by SlowFastMTB; and the number above the bounding box corresponds to the probability of the presence of smoke predicted by SlowFastMTB.

The performance of the proposed model was evaluated with the Video B dataset. As shown in Table 6, the confusion matrix presents that the true positive (TP), FP (type I error), FN (type II error), and true negative (TN) were 43.3%, 0%, 6.7%, and 50%, respectively. It is worth noting that the false alarm rate was zero. All errors were attributed to FN. The observed misclassification occurred when the size of smoke was smaller than 0.04% out of the image. When the amounts of the smoke become large, the proposed algorithm did not fail to detect the smoke. Three different metrics, including accuracy, recall, and precision, were used to further understand the performance of the proposed model. The accuracy is defined as $(TP + TN)/(TP + TN + FP + FN)$. The recall is defined as $TP/(TP + FN)$. The precision is defined as $TP/(TP + FP)$. The overall accuracy was found to be 93.3% in its ability to detect forest fires. The recall was 86.5%, while the precision was 100%.

Three existing deep learning models, including CNN, faster R-CNN, and SlowFast, were used for the purpose of comparison with the Video B dataset. The accuracy of CNN for detecting smoke was 50%. The visual images classified as "smoke" were further examined using Grad-CAM to identify whether smoke was really detected. The location of the smoke was visualized on a heat map. As depicted in Fig. 12a, CNN correctly classified the image as smoke. However, the Grad-CAM result indicates smoke in a wrong place. The same result was observed in another image that contains both smoke and the background mountain, as

Table 6: Confusion matrix of the proposed SlowFastMTB method.

		Actual result	
		True (smoke)	False (nonexistence of smoke)
Predicted result	True (smoke)	3504 (TP; 43.3%)	0 (FP; 0%)
	False (nonexistence of smoke)	546 (FN; 6.7%)	4050 (TN; 50.0%)

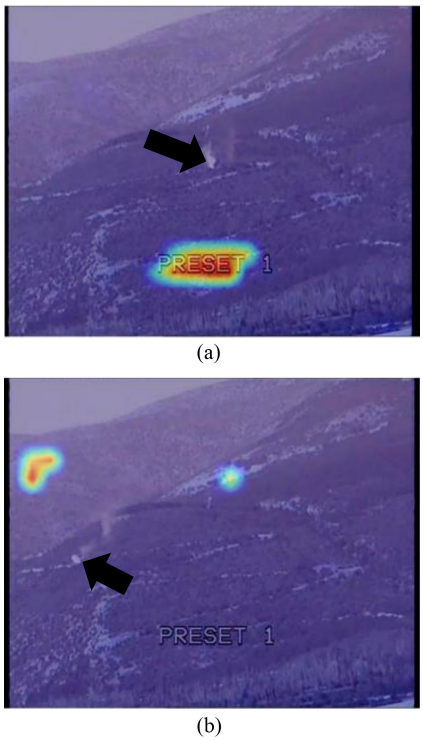


Figure 12: Representative results of CNN using the Video B dataset; (a) CNN classified the image as a smoke correctly. However, the Grad-CAM result showed that the smoke is in a wrong place and (b) CNN classified the image as a smoke correctly. However, the Grad-CAM result indicates the mountain are smoke.

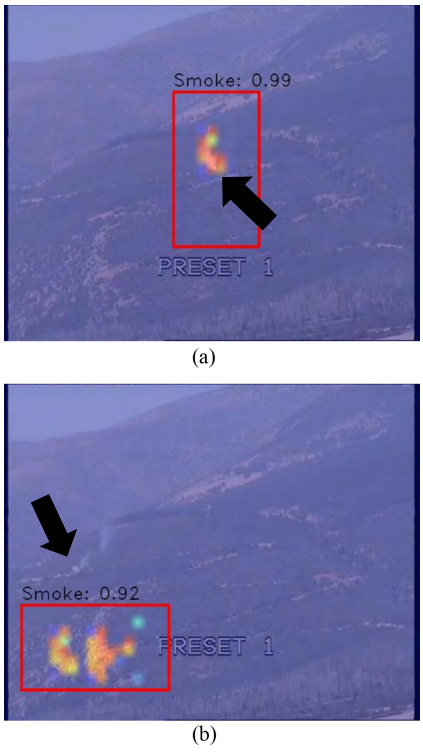


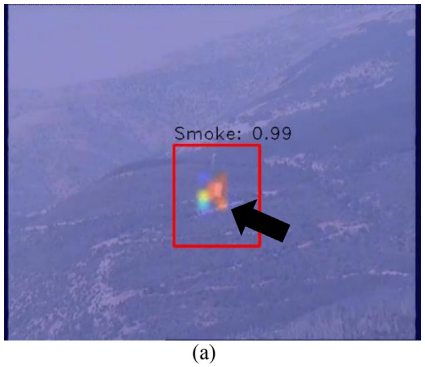
Figure 13: Representative results of faster R-CNN using the Video B dataset; (a) the smoke was detected correctly, and (b) the road in the bounding box is predicted as smoke incorrectly. The Grad-CAM also indicates the location of the actual smoke incorrectly.

presented in Fig. 12b. Therefore, it can be concluded that the actual accuracy of CNN is lower than 50%.

The accuracy of faster R-CNN for detecting smoke was 44.7%, which was lower than that of CNN. Nonetheless, it cannot be concluded that the performance of faster R-CNN was inferior to that of CNN. Faster R-CNN presents a bounding box. As shown in Fig. 13a, smoke was detected correctly with the correct placement of heat map. Another experiment in Fig. 13b showed that smoke was correctly detected. However, the heat map indicates the smoke as well as the road. This indicated that bounding box of faster R-CNN was not effective.

The accuracy of SlowFast for detecting smoke was 84.8%, which is much higher than both CNN and faster R-CNN. However, SlowFast often showed unstable results. As represented in Fig. 14, the smoke was detected correctly in Fig. 14a. However, as presented in Fig. 14b, it still shows the problem of focusing on the road side. This shows that the SlowFast method was not robust.

The performance results of the deep learning models with the Video B dataset are summarized in Table 7. The accuracy of CNN, faster R-CNN, SlowFast, and SlowFastMTB was 50.0%, 44.7%, 84.8%, and 93.3%, respectively. It was observed that the accuracy of SlowFastMTB was the highest (i.e., 93.3%), whereas that of faster R-CNN was the lowest (44.7%). As expected, the error rate ($= 1 -$ Accuracy) of faster R-CNN is the highest (55.3%), whereas that of SlowFastMTB is the lowest (6.7%). The errors can be divided into the FP rate and the FN rate. In particular, the FN rate is critical for evaluating the performance, since a missing alarm can lead to an evolution of a trivial forest fire to serious one. The FN rate of SlowFastMTB was only 6.7%, whereas the FN rates of CNN, faster R-CNN, and SlowFast were 40.0%, 13.9%, and 14.8%, respectively. F1 score was used to evaluate performance of the proposed model. The F1 score of CNN, faster R-CNN, SlowFast, and SlowFastMTB was 0.286, 0.566, 0.822, and 0.928, respectively. It was observed that the F1 score of SlowFastMTB was the highest (i.e., 0.928), whereas that of CNN was the lowest (i.e., 0.286). To verify the performance of object detection, the values of intersection over union (IOU) were incorporated. The IOU value of faster R-CNN, SlowFast, and SlowFastMTB was 0.440, 0.573, and 0.865, respectively. In the case of CNN, IOU values cannot be calculated since objective detection tasks cannot be carried out. It was observed that the IOU value of SlowFastMTB was the highest (i.e., 0.865), whereas that of faster R-CNN was the lowest (i.e., 0.440).



(a)

(b)

Figure 14: Representative results of SlowFast using the Video B dataset; (a) the smoke was detected correctly and (b) the road was incorrectly detected as smoke. The Grad-CAM also indicates a wrong place.

The SlowFastMTB model and CNN, faster R-CNN, and SlowFast trained with the Video B dataset were tested using videos C and D datasets. The performance of the models for Video C is summarized with the evaluation metrics including accuracy and IOU in Table 8. The accuracy of CNN, faster R-CNN, SlowFast, and SlowFastMTB was 52.0%, 85.0%, 97.5%, and 98.0%, respectively. It was observed that the accuracy of SlowFastMTB was the highest (i.e., 98.0%), whereas that of CNN was the lowest (52.0%). The IOU value of faster R-CNN, SlowFast, and SlowFastMTB was 0.739, 0.821, and 0.980, respectively. It was also observed that the IOU value of SlowFastMTB was the highest (i.e., 0.980), whereas that of faster R-CNN was the lowest (0.739). The difference between SlowFastMTB and SlowFast was small (i.e., only 0.5%) with the evaluation metric of accuracy. However, with that of IOU, the proposed SlowFastMTB outperformed SlowFast by 0.159.

The performance of the models for Video D is described in Table 9. The proposed SlowFastMTB model outperformed existing models in terms of accuracy and IOU. Therefore, it was concluded that the proposed SlowFastMTB model achieves enhanced detection accuracy, while reducing FN rates.

6. Conclusions and Future Work

This paper proposed a new deep learning model that detects smoke in forest fire videos for early forest fire detection. The proposed model combined the SlowFast method with a new annotation algorithm, namely MTB. In this study, the MTB algorithm was devised to isolate smoke with fuzzy boundaries from a background image. The performance of the proposed SlowFastMTB model was evaluated with a real forest fire video dataset recorded from the field. The accuracy, recall, and precision of the proposed model for the Video B dataset were 93.3%, 86.5%, and 100%, respectively. The proposed SlowFastMTB model outperformed existing models, including CNN, faster R-CNN, and SlowFast. The accuracy of SlowFastMTB was higher than CNN, faster R-CNN, and SlowFast by 43.3%, 48.6%, and 8.5%, respectively. The proposed model showed a significant reduction of FN rates. With additional datasets of videos C and D, it was demonstrated that the proposed SlowFastMTB model outperformed the existing models. The accuracy of SlowFastMTB with the Video C dataset was higher than CNN, faster R-CNN, and SlowFast by 46.0%, 13.0%, and 0.5%, respectively. The accuracy of SlowFastMTB with the Video D dataset was higher than CNN, faster R-CNN, and SlowFast by 17.0%, 9.0%, and 5.5%, respectively.

The outperformance of the proposed SlowFastMTB model over existing models is mainly attributed to the nature of video data. The SlowFastMTB approach captures the evolution of smoke over time from consecutive images of video clips. In contrast, image-based deep learning models extract information from a single image and ignore the correlation between consecutive images. The proposed SlowFastMTB also outperformed another video-based deep learning model (i.e., the original SlowFast model). Thus, it was concluded that the new MTB algorithm was effective for locating small amounts of smoke with fuzzy boundaries from background images. The MTB algorithm can also be used for automated annotation of other objects with ambiguous boundaries, such as flames and clouds. It should be

Table 7: Comparison of the proposed method with existing methods (Video B dataset).

	Metric	CNN	Faster R-CNN	SlowFast	SlowFastMTB (Proposed)
Error	Accuracy (%)	50.0	44.7	84.8	93.3
	FP rate (%)	10.0	41.5	0.4	0
	FN rate (%)	40.0	13.9	14.8	6.7
	F1 score	0.286	0.566	0.822	0.928
	IOU	N/A	0.440	0.573	0.865

Table 8: Comparison of the proposed method with existing methods (Video C dataset).

Metric	CNN	Faster R-CNN	SlowFast	SlowFastMTB (Proposed)
Accuracy (%)	52.0	85.0	97.5	98.0
IOU	N/A	0.739	0.821	0.980

Table 9: Comparison of the proposed method with existing methods (Video D dataset).

Metric	CNN	Faster R-CNN	SlowFast	SlowFastMTB (Proposed)
Accuracy (%)	60.5	68.5	72.0	77.5
IOU	N/A	0.578	0.680	0.775

noted that the SlowFastMTB model does not require human effort to annotate bounding boxes.

Two limitations of this study are presented for future work. First, the SlowFastMTB model for detecting small amounts of smoke from incipient forest fires can be used only when the movement of the smoke is faster than other objects in the background images. Further, if a moving smoke-like object (e.g., a white-colored train) also appears in the images studied, the accuracy of the SlowFastMTB model may be reduced. The SlowFastMTB model should be improved by incorporating a relevant algorithm (e.g., optical flow) to address this challenge. Second, the hyperparameters of the SlowFastMTB model should be optimized by leveraging advanced metaheuristic algorithms such as monarch butterfly optimization, earthworm optimization algorithm, elephant herding optimization, moth search algorithm, slime mould algorithm, and Harris hawks optimization. In this study, one of simple statistical tools, analysis of variance (ANOVA), was employed. This may not be desirable. If the hyperparameters are optimized in a systematic way, it is expected that the performance of the proposed SlowFastMTB model can be increased for detecting small amounts of smokes from incipient forest fires.

Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2021R1A2C1008143).

Conflict of interest statement

All the authors declare that there is no actual or potential conflict of interest including any financial, personal or other relationships with other people or organizations.

References

Alkhatib, A. A. (2014). A review on forest fire detection techniques. *International Journal of Distributed Sensor Networks*, *10*, 597368.

Burić, M., Pobar, M., & Ivašić-Kos, M. (2018). Object detection in sports videos. In *Proceedings of the 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)* (pp. 1034–1039).

Cetin, A. E. *Sample fire and smoke video clips [Online]*. Available: http://signal.ee.bilkent.edu.tr/VisiFire/Demo/ForestSmoke/. Accessed on April 6, 2022.

Chen, G., Hay, G. J., Carvalho, L. M. T., & Wulder, M. A. (2012). Object-based change detection. *International Journal of Remote Sensing*, *33*, 4434–4457.

Chen, J., You, Y., & Peng, Q. (2013). Dynamic analysis for video based smoke detection. *International Journal of Computer Science Issues*, *10*, 298–304.

Coop, J. D. et al. (2020). Wildfire-driven forest conversion in Western North American landscapes. *Bioscience*, *70*, 659–673.

Cui, Z., Xue, F., Cai, X., Cao, Y., Wang, G.-g., & Chen, J. (2018). Detection of malicious code variants based on deep learning. *IEEE Transactions on Industrial Informatics*, *14*, 3187–3196.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 17th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (pp. 4171–4186).

Eiben, A. E., & Smith, J. E. (2015). *Introduction to evolutionary computing*. Springer Publishing Company, Inc.

Feichtenhofer, C., Fan, H., Malik, J., & He, K. (2019). Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 6202–6211).

Feng, Y., Deb, S., Wang, G.-G., & Alavi, A. H. (2021). Monarch butterfly optimization: A comprehensive review. *Expert Systems with Applications*, *168*. https://doi.org/10.1016/j.eswa.2020.114418.

Gao, D., Wang, G. G., & Pedrycz, W. (2020). Solving fuzzy job-shop scheduling problem using DE algorithm improved by a selection mechanism. *IEEE Transactions on Fuzzy Systems*, *28*, 3265–3275.

Ghali, R., Akhloufi, M. A., Jmal, M., Mseddi, W. S., & Attia, R. (2021). Wildfire segmentation using deep vision transformers. *Remote Sensing*, *13*, 7.

Goncalves, A., Ray, P., Soper, B., Stevens, J., Coyle, L., & Sales, A. P. (2020). Generation and evaluation of synthetic patient data. *BMC Medical Research Methodology*, *20*, 1–40.

Gong, J., Yue, Y., Zhu, J., Wen, Y., Li, Y., Zhou, J., Wang, D., & Yu, C. (2012). Impacts of the Wenchuan earthquake on the Chaping river upstream channel change. *International Journal of Remote Sensing*, *33*, 3907–3929.

Han, D., & Lee, B. (2006). Development of early tunnel fire detection algorithm using the image processing. In *Proceedings of the International Symposium on Visual Computing* (pp. 39–48).

He, K., Zhang, X., Ren, S., & Sun, J. (2016a). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 770–778).

He, K., Zhang, X., Ren, S., & Sun, J. (2016b). Identity mappings in deep residual networks. In *Proceedings of the European Conference on Computer Vision* (pp. 630–645).

Healey, S. P. et al. (2018). Mapping forest change using stacked generalization: An ensemble approach. *Remote Sensing of Environment*, *204*, 717–728.

Jiao, Z., Zhang, Y., Xin, J., Mu, L., Yi, Y., Liu, H., & Liu, D. (2019). A deep learning based forest fire detection approach using UAV and YOLOv3. In *Proceedings of the 2019 1st International Conference on Industrial Artificial Intelligence (IAI)* (pp. 1–5).

Kim, B., & Lee, J. (2019). A video-based fire detection using deep learning models. *Applied Sciences*, *9*, 4.

Kim, C., Oh, H., Jung, B. C, & Moon, S. J. (2021). Optimal sensor placement to detect ruptures in pipeline systems subject to uncertainty using an Adam-mutated genetic algorithm. *Structural health monitoring*. online published.

Ko, B. *KMU fire and smoke database [Online]*. Available: https://cvpr.kmu.ac.kr/Dataset/Wildfire_smoke.zip. Accessed on April 6, 2022.

Koo, B., & Shin, B. (2018). Applying novelty detection to identify model element to IFC class misclassifications on architectural and infrastructure building information models. *Journal of Computational Design and Engineering*, 5, 391–400.

Lee, C.-Y., Lin, C.-T., & Hong, C.-T. (2009). Spatio-temporal analysis in smoke detection. In *Proceedings of the IEEE International Conference on Signal and Image Processing Applications* (pp. 80–83).

Li, M., Xu, W., Xu, K., Fan, J., & Hou, D. (2013). Review of fire detection technologies based on video image. *Journal of Theoretical and Applied Information Technology*, 49, 700–707.

Li, M., Wang, G.-G., & Yu, H. (2021). Sorting-based discrete artificial bee colony algorithm for solving fuzzy hybrid flow shop green scheduling problem. *Mathematics*, 9, 8.

Lin, G., Zhang, Y., Xu, G., & Zhang, Q. (2019). Smoke detection on video sequences using 3D convolutional neural networks. *Fire Technology*, 55, 1827–1847.

Liu, T., Yang, L., & Lunga, D. (2021). Change detection using deep learning approach with object-based image analysis. *Remote Sensing of Environment*, 256. https://doi.org/10.1016/j.rse.2021.112308.

Oh, H., Jung, J. H., Jeon, B. C., & Youn, B. D. (2018). Scalable and unsupervised feature engineering using vibration-imaging and deep learning for rotor system diagnosis. *IEEE Transactions on Industrial Electronics*, 65, 3539–3549.

Park, J., Ko, B., Nam, J.-Y., & Kwak, S. (2013). Wildfire smoke detection using spatiotemporal bag-of-features of smoke. In *Proceedings of the 2013 IEEE Workshop on Applications of Computer Vision* (pp. 200–205).

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems*, 28, 91–99.

Shukla, B. P., & Pal, P. (2009). Automatic smoke detection using satellite imagery: Preparatory to smoke detection from INSAT-3D. *International Journal of Remote Sensing*, 30, 9–22.

Siscawati, M. (1998). Underlying causes of deforestation and forest degradation in Indonesia: A case study on forest fire. In *Proceedings of the IGES International Workshop on Forest Conservation Strategies for the Asia and Pacific Region* (pp. 44–57).

Starr, J. W., & Lattimer, B. Y. (2012). A comparison of IR stereo vision and LIDAR for use in fire environments. In *Proceedings of the 2012 IEEE Sensors* (pp. 1–4).

Sun, X., Sun, L., & Huang, Y. (2021). Forest fire smoke recognition based on convolutional neural network. *Journal of Forestry Research*, 32, 1921–1927.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1–9).

Tan, M., & Le, Q. (2019). EfficientNet: Rethinking model scaling for convolutional neural networks. In *Proceedings of the 36th International Conference on Machine Learning* (pp. 6105–6114).

Tang, Z., Liu, X., Chen, H., Hupy, J., & Yang, B. (2020). Deep learning based wildfire event object detection from 4K aerial images acquired by UAS. *AI*, 1, 166–179.

Thomas, P. J., & Nixon, O. (1993). Near-infrared forest fire detection concept. *Applied Optics*, 32, 5348–5355.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. In *Proceedings of the 31st Conference on Neural Information Processing Systems*.

Wang, G., Guo, L., & Duan, H. (2013). Wavelet neural network using multiple wavelet functions in target threat assessment. *Scientific World Journal*, 2013, 632437.

Wang, G.-G., Lu, M., Dong, Y.-Q., & Zhao, X.-J. (2016). Self-adaptive extreme learning machine. *Neural Computing and Applications*, 27, 291–303.

Wang, S., Ma, Q., Ding, H., & Liang, H. (2018). Detection of urban expansion and land surface temperature change using multi-temporal Landsat images. *Resources, Conservation and Recycling*, 128, 526–534.

Williams, A. P., Abatzoglou, J. T., Gershunov, A., Guzman-Morales, J., Bishop, D. A., Balch, J. K., & Lettenmaier, D. P. (2019). Observed impacts of anthropogenic climate change on wildfire in California. *Earth's Future*, 7, 892–910.

Xiong, Z., Caballero, R., Wang, H., Finn, A.M., Lelic, M. A., & Peng, P.-Y. (2007). Video-based smoke detection: Possibilities, techniques, and challenges. In *Proceedings of the IFPA Fire Suppression and Detection Research and Applications—A Technical Working Conference (SUPDET)*.

Xu, Z., & Xu, J. (2007) Automatic fire smoke detection based on image visual features. In *Proceedings of International Conference on Computational Intelligence and Security Workshops* (pp. 316–319).

Xu, G., Zhang, Q., Liu, D., Lin, G., Wang, J., & Zhang, Y. (2019). Adversarial adaptation from synthesis to reality in fast detector for smoke detection. *IEEE Access*, 7, 29471–29483.

Yi, J.-H., Wang, J., & Wang, G.-G. (2016). Improved probabilistic neural networks with self-adaptive strategies for transformer fault diagnosis problem. *Advances in Mechanical Engineering*, 8, pp. 1–13.

Yuan, F., Fang, Z., Wu, S., Yang, Y., & Fang, Y. (2015). Real-time image smoke detection using staircase searching-based dual threshold adaboost and dynamic analysis. *IET Image Processing*, 9, 849–856.

Yuan, F., Zhang, L., Xia, X., Wan, B., Huang, Q., & Li, X. (2019). Deep smoke segmentation. *Neurocomputing*, 357, 248–260.

Zhang, Q.-x., Lin, G.-h., Zhang, Y.-m., Xu, G., & Wang, J.-j. (2018). Wildland forest fire smoke detection based on faster R-CNN using synthetic smoke images. *Procedia Engineering*, 211, 441–446.