IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received April 5, 2022, accepted June 14, 2022, date of publication June 21, 2022, date of current version June 24, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3184707

Forest-Fire Response System Using Deep-Learning-Based Approaches With CCTV Images and Weather Data

DAI QUOC TRAN[1], MINSOO PARK[1], YUNTAE JEON[1], JINYEONG BAK[2], AND SEUNGHEE PARK[1,3]

[1]School of Civil, Architectural Engineering and Landscape Architecture, Sungkyunkwan University, Suwon 16419, South Korea
[2]College of Computing and Informatics, Sungkyunkwan University, Suwon 16419, South Korea
[3]Technical Research Center, Smart Inside AI Company Ltd., Suwon 16419, South Korea

Corresponding author: Seunghee Park (shparkpc@skku.edu)

This work was supported in part by the Ministry of Interior and Safety (MOIS)'s Project for the development of accident prevention technology for vulnerable groups under Grant 2022-MOIS38-002; and in part by the National Research Foundation of Korea (NRF) through the Ministry of Science and ICT (MSIT), Korea Government, under Grant NRF-2021R1A4A3033128.

ABSTRACT An effective forest-fire response is critical for minimizing the losses caused by forest fires. The purpose of this study is to construct a model for early fire detection and damage area estimation for response systems based on deep learning. First, we implement neural architecture search-based object detection (DetNAS) for searching optimal backbone. Backbone networks play a crucial role in the application of deep learning-based models, as they have a significant impact on the performance of the model. A large-scale fire dataset with approximately 400,000 images is used to train and test object-detection models. Then, the searched light-weight backbone is compared with well-known backbones, such as ResNet, VoVNet, and FBNetV3. In addition, we propose damage area estimation method using Bayesian neural network (BNN), data pertaining to six years of historical forest fire events are employed to estimate the damaged area. Subsequently, a weather API is used to match the recorded events. A BNN model is used as a regression model to estimate the damaged area. Additionally, the trained model is compared with other widely used regression models, such as decision trees and neural networks. The Faster R-CNN with a searched backbone achieves a mean average precision of 27.9 on 40,000 testing images, outperforming existing backbones. Compared with other regression models, the BNN estimates the damage area with less error and increased generalization. Thus, both proposed models demonstrate their robustness and suitability for implementation in real-world systems.

INDEX TERMS Forest-fire management, deep learning, Bayesian neural network, object detection.

I. INTRODUCTION

Forests contribute to significant ecological and economic functions in ecosystems [1]. In addition, forests are important heritage sites for human beings. However, forest fires can cause tremendous damage to human life and property and adversely affect forest ecosystems in the long term. Therefore, forest fires must be prevented; however, they are difficult to prevent because of their diverse causes. According to the Republic of Korea's Forest Fire Statistics Yearbook [2], the risk of forest fires increased in 2020 owing to an increase in the number of dry days and a considerable decrease in the number of precipitation days. A total of 620 forest fires occurred in 2020, causing damage over an area of 2919.84 hectares. The number of forest fires increased by 31% in the last decade (474 cases), and the area of forest being damaged increased by 161% (1120 hectares). The initial response to forest fires is an important factor in reducing accidents. Thus, comprehensive real-time monitoring and damage assessment are necessary for minimizing forest-fire-related losses.

Owing to the advancement of deep-learning-based vision approaches, the limitations of traditional methods for classification [3]–[7], object detection [8]–[14], and segmentation [15]–[17] can be overcome. Object-detection models that exhibit high performance can be categorized into two types,

The associate editor coordinating the review of this manuscript and approving it for publication was Farid Boussaid.

based on the model structure. The first type is one-stage detection models, such as the You Only Look Once (YOLO) series [18], single-shot detectors [19], and RetinaNet [20]. The second type is two-stage detection models, such as faster region-based convolutional neural networks (Faster R-CNNs) [21], DenseNet [22], and Mask R-CNN [23]. In one-stage models, regional proposals and classifications are performed simultaneously in one step. For example, some researchers [24] used YOLOv3 [25] with unmanned aerial vehicle (UAV) images to detect fires and smoke. This model achieved a recognition rate and frame rate of approximately 8% and 3.2 frames per second (fps), respectively. Other researchers [26] combined two one-stage models, Yolov5 and EfficientDet [9], for fire detection. The ensemble model produced an average precision (AP) of 0.79 and latency of 66.8 ms. However, these single-stage models were insufficient for detecting forest fires covering large areas. Therefore, we adopt two-stage models in which regional proposal and classification are performed sequentially in two steps. Two-stage models, such as ATT Squeeze U-Net [11], which was recently developed, obtained an accuracy of 0.93 and detection frame rate of 1.1 fps (0.89 s per image). In addition, the Faster R-CNN was used with three different base networks, AlexNet, VGG16, and ResNet101 [8], for forest-fire detection. Researchers [27] used the Faster R-CNN with data from a real monitoring system; the results showed that the trained model can achieve an F1-score of approximately 80%. Recently, an efficient convolutional neural network (CNN) architecture was developed for classifying input images into eight distinct fire-scenario classes [28]. However, these models have several limitations.

1) Nonreasonable Hyperparameters: The original Faster R-CNN models based on various backbones, such as ResNet [29] and VoVNet [30], have a problem with training hyperparameters set based on the experiences and intuitions of the researchers. This negatively affects the accuracy and speed of the model.
2) The searched backbone, such as FBNetV3 [31], may fall into a sub-optimal point when trained on an object-detection dataset.
3) Low Inference Rate: Two-stage detection models have an essential problem with slowness. For a real-time forest-fire detection system, both the detection accuracy and rate are important.

In this study, DetNAS [32] is used to search for an optimal backbone for the smoke and forest-fire detection model. The NAS-based [33] searching backbone algorithm is considered as a strong tool for finding an optimal CNN architecture suitable for domain datasets. ShuffleNet V2 [34], an efficient light-weight architecture, is used for searchable components. A publicly available large-scale fire dataset of approximately 350,000 images is used for training. The searched backbone is evaluated on approximately 40,000 images and compared with the original Faster R-CNN models using the well-known ResNet [29], light-weight VoVNet [30] and NARS-based FBNetV3 backbones [31].

Estimating the damage area is critical for providing the appropriate response. According to [2], forest-fire response scenarios can be of four levels. The first is the early response scenario, which represents the case when the estimated damage area is less than 10 hectares, fire duration is less than 3 hours, and wind speed is less than 2 m/s. In this case, local authorities utilize 50% of their firefighters and equipment and 100% of their helicopters. The second step is the scenario in which the estimated burned area is between 10 and 30 hectares, duration is between 3 and 8 hours, and wind speed is between 2 and 4 m/s. In this scenario, all firefighters, equipment, and helicopters are used, along with 50% of the helicopter and drone capabilities from neighboring provinces, boosting the response performance. The third is the scenario in which the estimated damage area is between 30 and 100 hectares, duration is between 8 and 24 hours, and wind speed is between 4 and 7 m/s. In this case, 50% of firefighting personnel, 30% of equipment, and 100% of helicopter and drone resources from neighboring provinces are used. The fourth is the scenario in which the estimated burned area exceeds 100 hectares, duration exceeds 24 hours, and wind speed is greater than 7 m/s. In this case, 50% of equipment is acquired from surrounding provinces, helicopters are acquired from metropolitan areas, and all other facilities are acquired from local governments.

Fire characteristics that are influenced by environmental conditions, such as wind speed, air temperature, humidity, ground temperature, and atmospheric pressure, affect the burned area during a forest fire. Thus, understanding the relationship between the damaged area, its components, and the available manpower factors (firefighter and helicopter involvement) is critical for providing the best possible response to these natural disasters. To account for the interdisciplinary domain knowledge necessary to estimate the damage area, model frameworks that can consider the interdependencies between the processes involved are required. Bayesian neural networks (BNNs) are ideal for combining multidisciplinary models [35]. Bayesian networks (BNs) were applied in previous studies on tsunamis [36] and rock fall hazards [35]. Researchers [37] used a BNN to predict and assess wildfire occurrences and burn severity and modeled the wildfire spread [38], wildfire ecological consequences [39], and risk of human fatality from fires in buildings [40]. In 2017, an approach that utilized BNs for wildfire economic losses was presented [41]. BNs were recently used to forecast and analyze the causes of forest fires [42]. BNs have been employed in various studies. Owing to the complexity and the involvement of many factors in forest fires, assessing forest-fire damage is currently problematic. In this study, historical forest-fire data were used to establish a baseline for implementing a BNN. Weather and response variables were included in the input data. The BNN algorithm was also evaluated and compared with other machine-learning algorithms.

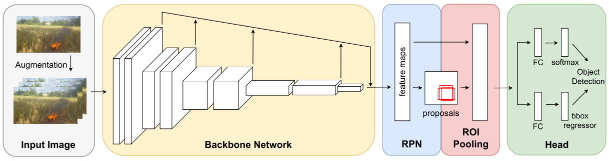


FIGURE 1. Faster R-CNN architecture.

To summarize, the main contributions of this study are as follows:

1) Propose an approach for forest-fire response systems by using deep learning.
2) Search for optimal light-weight backbones on a large-scale fire and smoke dataset.
3) Implement a BNN for estimating the damage area using six years of forest-fire records and weather data.

The remainder of this paper is organized as follows. Section II provides an outline of the theoretical background. Section III explains the proposed system. Section IV describes our experiments. Section V presents a summary of the findings. Finally, Section VI highlights the outcomes.

II. THEORETICAL BACKGROUND

A. DetNAS-BASED OBJECT DETECTION

The method for detecting forest fires using closed-circuit television (CCTV) footage is primarily based on object detection. Figure 1 shows the architecture of the Faster R-CNN. The model receives an image as the input, and the image then passes through the backbone of the model. After convolution in the backbone, the feature maps of the image are extracted. Then, a region proposal network (RPN) is applied to the feature maps. The output values of the RPN are proposals, such as the red boxes in the RPN. These proposals and feature maps from the backbone are used to perform a region-of-interest pooling to create a fully connected layer. Finally, object detection is completed using softmax and bounding box regression. Object detectors rely heavily on the backbone [10]. However, the backbone, which is hand-crafted and optimized for image classification datasets, is frequently applied directly to the object detection model. This operation may have a negative impact on performance [43]. Recent advances in deep learning have resulted in state-of-the-art models in various fields by using neural architecture search (NAS) [33]. NAS was intended to address the issues associated with early models that occur when researchers build models based on their experience. NAS enabled the creation of a model architecture without human interaction.

Many studies on backbones and hyperparameters were conducted to improve the performance of object detection models. In the case of MobileNetV2 (Fig. 2.(a)), both the backbone network and parameters were determined by many experiments, not by NAS. Models based on human experience are potentially problematic in terms of rate and accuracy. In the case of FBNet-C (Fig. 2.(b)), deep learning creates the backbone network, but the hyperparameters are also determined experimentally. However, in the case of FBNetV3 [31] (Fig 2.(c)), deep learning determines both the backbone and hyperparameters. Fig. 2 shows the backbone and expansion values of each block. The expansion values of MobileNetV2 and FBNet-C have 3 or 6 that are fixed by a human. However, the expansion values of FBNetV3 are continuously distributed using deep learning. Consequently, FBNetV3 outperforms other networks [31].

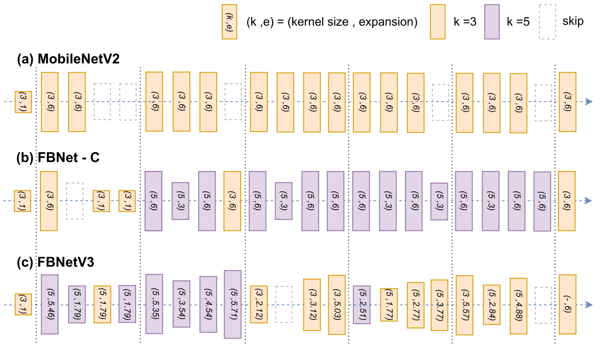


FIGURE 2. Visualization of backbone networks. The rectangular boxes represent blocks for each layer. We used two colors to represent the kernel size of each layer; orange - 3, purple - 5, and blank indicates a skip [31], [44].

Numerous studies were conducted for automatically determining backbones, but a limits exist as to what can be applied for forest fire detection, including FBNetV3. The majority of previous studies attempted to identify a backbone network for image classification by using ImageNet, a dataset for image classification. They then attempted to apply these image-classification backbones to an object detection task. Because of this mismatch, performance is limited. DetNAS [43] is the first study to investigate NAS for determining the object detection backbone. The researchers applied a single path one-shot method [45] for NAS and split the search process into 3-steps: supernet pretraining, supernet fine-tuning, and an evolution search on the trained supernet.

Figure 3 shows each step in NAS for searching for a forest fire detection backbone. First, supernet pretraining is determining the supernet, which is a set of many subnets. The supernet is trained using ImageNet. In this step, a path-wise method is applied for ensuring the relationship between the supernet and subnets. Second, the supernet, which includes a head and metrics, is fine-tuned. This supernet fine-tuning is trained using the coco-dataset, which is an object detection dataset. To customize the backbone, we must change the detection dataset and hyperparameters. Third, determine one subnet from the supernet. During this search, an evolutionary algorithm is used to select a candidate in the supernet.

B. BAYESIAN NEURAL NETWORK

As discussed earlier, the final aim was to build a model capable of estimating the damage area of forest fires based on current weather conditions and historical weather records. Therefore, the damaged areas were divided into classes, each

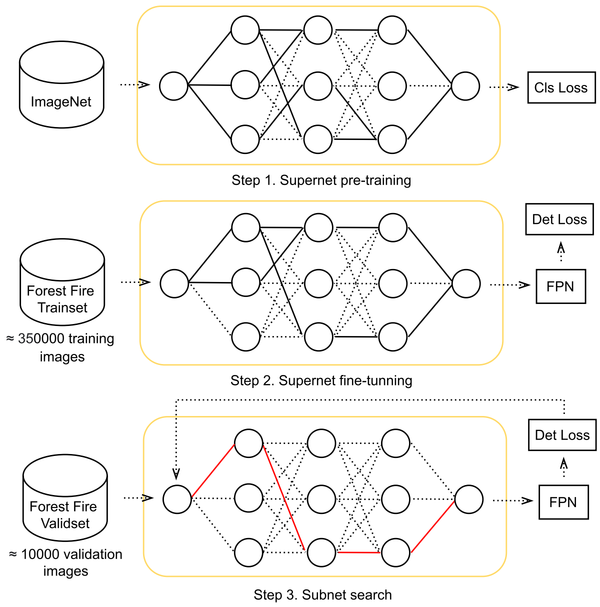


FIGURE 3. NAS process for object detection backbone [43].

with increasing values. Consequently, the problem can be classified as a regression problem from a machine-learning perspective. An artificial neural network (ANN) is a powerful tool for classification and regression problems [46]. This point-estimate approach, on the contrary, tends to lack reasonability and may generalize in an unexpected and overconfident manner on data points from outside the training distribution [47]. As mentioned earlier, BNNs are widely used in risk-assessment fields. A BNN is a common type of stochastic neural network, which can better understand the uncertainty of underlying processes [48]. Figure 4 shows the distinction between a point-estimate neural network and BNN, that is, instead of adopting fixed numbers as model weights, a distribution represents each weight in the BNN.

A BNN can be summarized as follows:

$$\theta \sim p(\theta), \tag{1}$$

$$y = NN_\theta(x) + \varepsilon, \tag{2}$$

where θ represents the model parameters, and ε represents random noise. The following step selects a neural network architecture. $p(\theta)$ and $p(y|x, \theta)$ are a prior distribution over the possible model parameters, and the prior confidence in the predictive power of the model must be selected. Thus, the Bayesian posterior can be written as

$$p(\theta|D) = \frac{p(D_y|D_x, \theta)p(\theta)}{\int_\theta p(D_y|D_x, \theta')p(\theta')d\theta'} \propto p(D_y|D_x, \theta)p(\theta), \tag{3}$$

where D, D_x, and D_y are the training set, training features, and training label, respectively. In practice, computing this distribution is typically difficult. Thus, the sampling method (Markov chain Monte Carlo) and variational inference approach are applied as approximation methods. Refer to this reference for a more detailed discussion of these techniques [48]. The TensorFlow probabilistic deep-learning framework is used in this study to implement a BNN.

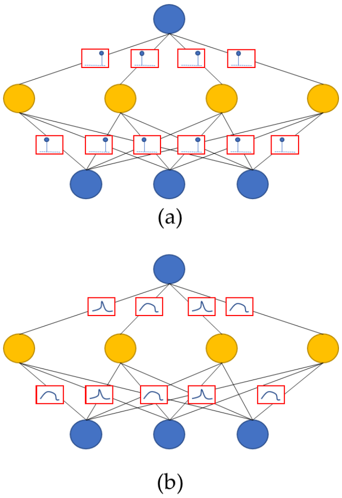


FIGURE 4. Point-estimate neural network (a) and stochastic coefficient neural networks (b); weights are learned as a probability distribution [48].

III. PROPOSED METHOD

The four phases of forest-fire management [49] are mitigation, preparation, response, and recovery. The focus of this study is on the implementation of a forest-fire response system with two primary aspects: detection and damage estimation. As shown in Figure 5, if smoke or fire occurs, AI-powered CCTV cameras detect smoke and fire in the forest in real time and then send the results to a database server. Subsequently, the server sends a UAV to the forest fire location to scan for damage. Then, the regression model gathers data on the extent of the damage area and weather data from the fire location to estimate and visualize it in an integrated system.

The most critical aspect of a forest-fire response system is the rapid and accurate detection. To achieve this, we propose a novel forest fire detection backbone network derived from NAS. Previous research on forest fire detection has relied exclusively on various object detection models that perform well on the COCO dataset [50], not on a forest fire detection dataset. Because our backbone is tailored to the large-scale forest fire dataset, our detection model with a searched backbone outperforms other fire detection models. Additionally, ShuffleNetV2 block is used as a searchable component for searching for the backbone. Therefore, the searched backbone can be considered as a light-weight model capable of real-time inference and deployable on edge devices [34].

After a forest fire is detected early, the next step is to estimate the damage area in real time. To estimate the damage area, a BNN-based regression model is constructed using historical forest fire records and weather data. BNNs are well-known for handling uncertainty in datasets. Consequently, the trained model can generate a distribution representing the probability of the damage area. Additionally,

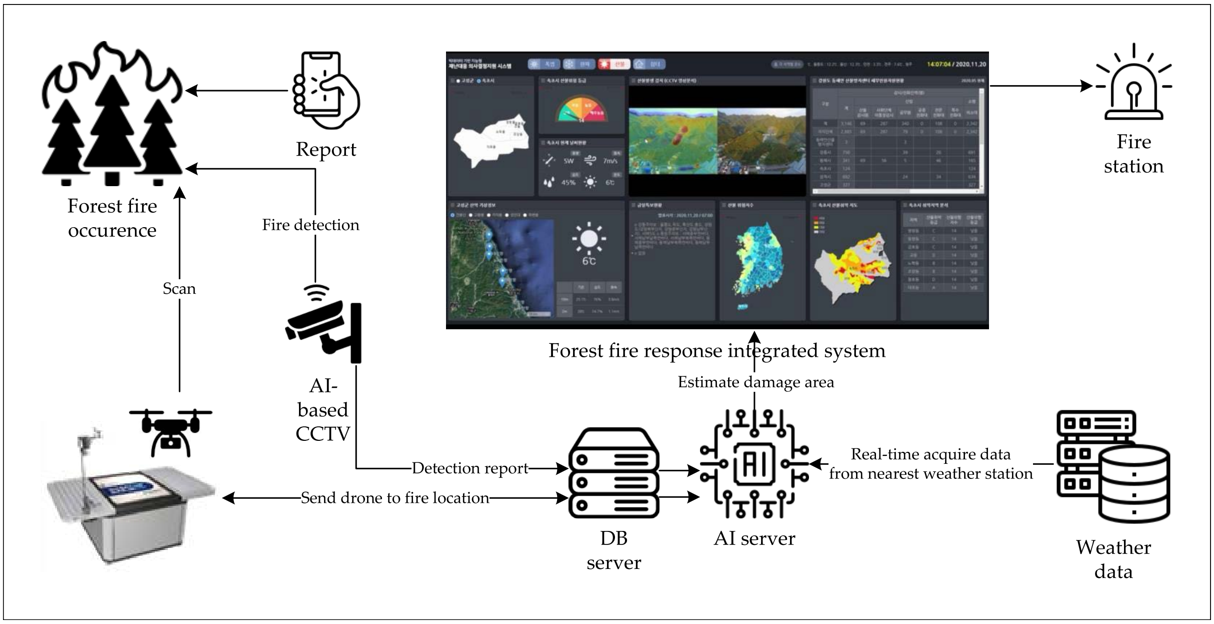


FIGURE 5. **Proposed deep-learning-based forest fire management system.**

using a UAV and segmentation algorithm, we could estimate the total damage area and create a 3D forest fire damage map using our previous research [17].

IV. EXPERIMENTS

In this study, we proposed to perform forest fire detection and damage area estimation to establish a forest-fire response system. We conducted two experiments, based on each proposal. For **forest fire detection**, we prepared a fire image dataset for training. In this process, we analyzed and preprocessed the training dataset to solve the unbalancing problem between classes. The DetNAS algorithm was then used to search for an optimal backbone. In addition, we compared our model with ResNet, VoVNet, and NAS-based FBNetV3 models. For **damage area estimation**, we matched the weather API to collect the time-series numerical dataset. Subsequently, we trained and validated our BNN-based damage area estimation model.

A. FOREST FIRE DETECTION

1) DATASET

The AIHub[1] fire detection dataset was used in the experiments. As shown in Figure 6a), 349774 images were used to train the model, and the remaining 39243 images were used to evaluate its performance. The number of instances per class from the training and testing datasets is shown in Fig. 6b). This dataset was divided into four distinct categories: black smoke, gray smoke, white smoke, and fire.

Figure 7 depicts the data sample with its corresponding classes. The data were created in a naturalistic environment, near a mountain, similar to a forest fire. Consequently, this dataset was well-suited for developing a model for the early detection of forest fires using CCTV data.

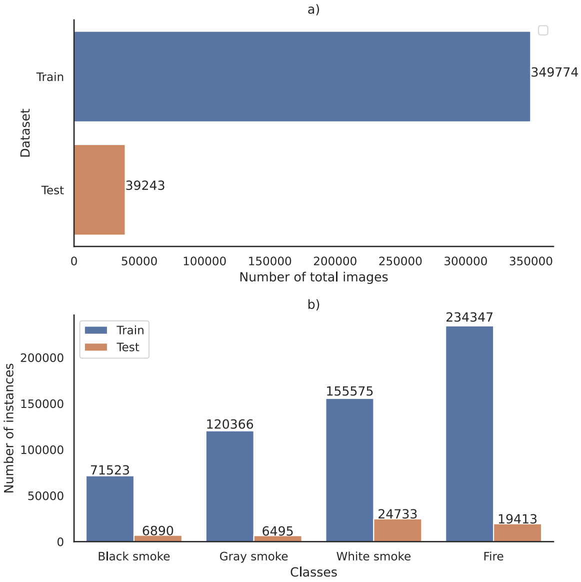


FIGURE 6. **Public dataset distribution. (a) Total number of training and test images. (b) Number of instances per class for training and test images.**

2) PREPROCESSING

We employed image augmentation techniques, such as random rotation, vertical and horizontal flipping, and their associated labeling for two reasons. First was the irregularity in the smoke shape. Figure 7) shows smoke with various shapes. Unlike fixed-shape objects, such as people and cars, smoke

[1] https://aihub.or.kr/aidata/34121

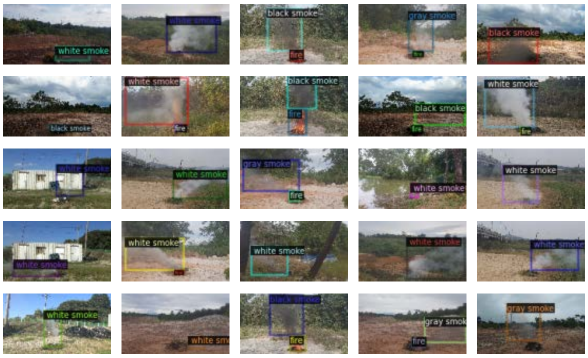


FIGURE 7. Examples of images with their associated labels.

adopts various shapes and directions. Therefore, image augmentation can be used effectively for training data augmentation because smoke is not a fixed-shape object. Second was the unbalanced distribution of the training dataset for each class. Figure 6.(b) shows that the number of instances for each class is uneven. To solve this, we performed image augmentation with different numbers of instances depending on each class. Using image augmentation in this manner improves the reasonability of the detection model.

3) METRICS

The COCO AP (average precision) was used to evaluate the model. The COCO AP is computed from the precision-recall (PR) curve based on true positive (TP), false positive (FP), and false negative (FN) results. Because our goal is to detect fire and not classify the colors of smoke or fire, we use the mean average precision (mAP), which is the most commonly used object detection model metric [10].

4) IMPLEMENTATION DETAILS

For the model architecture, we used the Faster R-CNN model with ResNet, VoVNet, FBNetV3 and our model with the searched backbone. The network with a feature pyramid network (FPN) attached to its name has a modified structure that creates feature maps step-by-step in the existing network layer, combines features in top-to-bottom manner, and proceeds with object detection. For training configurations, the batch size was 16, number of iterations was 10K, and initial learning rate was 0.15. The input image was resized to 512 × 512 pixels without changing the aspect ratio using bilinear interpolation.

In our experiment for NAS, as shown in Figure 3, the ImageNet dataset was used in step 1, nearly 350000 images were used to fine-tune the supernet, and 10000 images were used to search the subnet. We used an evolution algorithm to determine the best subnet from the candidates. During the evolutionary search, we used a candidate pool size of 50 and mutation number of 20. After determining the subnet, we retrained the Faster R-CNN model using the subnet on ImageNet and fine-tuned on 350000 forest fire images. All searching processes ran based on the MMRazor Opensource library [51]. In all steps, we used a 512 × 512 image-size to compare backbones. The input image was normalized and augmented with a random flip ($p = 0.5$). Stochastic gradient descent, using a learning rate ($lr = 0.001$), was used as an optimizer. Other configurations were adopted from the original research [43]. The models were implemented in Pytorch and ran on CentOS Linux 8 with two NVIDIA Tesla V100 graphics processor units, each with 32 GB of memory. Approximately 8 days were required for the two GPUs to train and fine-tune all DetNAS processes.

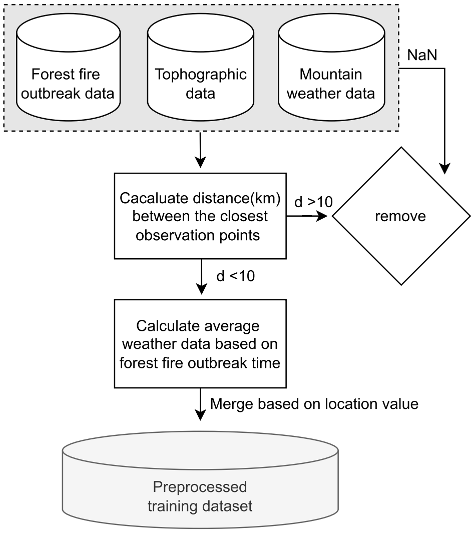

Forest fire outbreak data
Tophographic data
Mountain weather data
NaN
Cacaluate distance(km) between the closest observation points
d >10
remove
d <10
Calculate average weather data based on forest fire outbreak time
Merge based on location value
Preprocessed training dataset

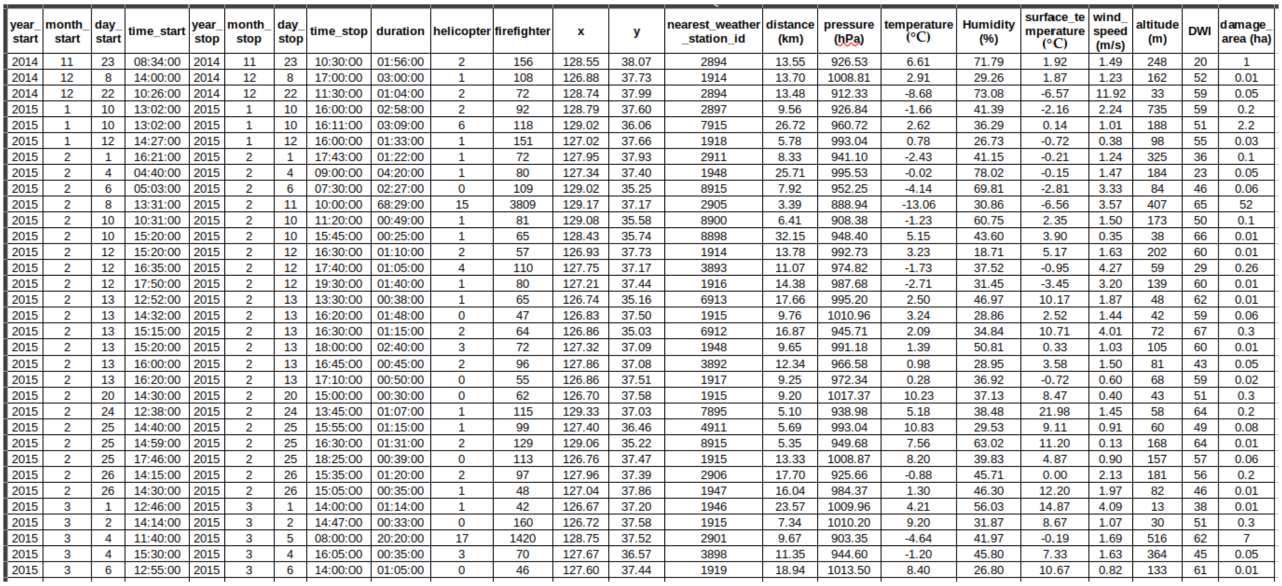
FIGURE 8. Matching process.

B. DAMAGE AREA ESTIMATION

1) DATASET

In this study, we used three datasets from the Korea Forest Service Organization [2]. The first is a forest fire outbreak dataset that includes a record of 2,128 forest fires from 2014 to 2020. The second is a topographic dataset, and the last is a mountain weather dataset from the 365-weather observation source API. Because these three datasets have different observation points, preprocessing should be implemented. Figure 8 shows the preprocessing. First, we discarded the rows with NaN and then integrated the three datasets based on the distances between the closest observation points. If the nearest event distance is less than 10 km, we regard it as having the same event point. After filtering, for the integration of fire outbreaks and weather data, we averaged the weather data based on the fire outbreak time and duration. We used 11 variables, as shown in Figure 9, namely, the month, duration, number of helicopters involved, number of firefighters, temperature, surface temperature, pressure, humidity, wind speed, altitude, and daily weather index (DWI).

The raw dataset contains all categories with the corresponding ranges shown in Table 1. This procedure was



year_ start	month_ start	day_ start	time_start	year_ stop	month_ stop	day_ stop	time_stop	duration	helicopter	firefighter	x	y	nearest_weather _station_id	distance (km)	pressure (hPa)	temperature (°C)	Humidity (%)	surface_te mperature (°C)	wind_ speed (m/s)	altitude (m)	DWI	damage_ area (ha)
2014	11	23	08:34:00	2014	11	23	10:30:00	01:56:00	2	156	128.55	38.07	2894	13.55	926.53	6.61	71.79	1.92	1.49	248	20	1
2014	12	8	14:00:00	2014	12	8	17:00:00	03:00:00	1	108	126.88	37.73	1914	13.70	1008.81	2.91	29.26	1.87	1.23	162	52	0.01
2014	12	22	10:26:00	2014	12	22	11:30:00	01:04:00	2	72	128.74	37.99	2894	13.48	912.33	-8.68	73.08	-6.57	11.92	33	59	0.05
2015	1	10	13:02:00	2015	1	10	16:00:00	02:58:00	2	92	128.79	37.60	2897	9.56	926.84	-1.66	41.39	-2.16	2.24	735	59	0.2
2015	1	10	13:02:00	2015	1	10	16:11:00	03:09:00	6	118	129.02	36.06	7915	26.72	960.72	2.62	36.29	0.14	1.01	188	51	2.2
2015	1	12	14:27:00	2015	1	12	16:00:00	01:33:00	1	151	127.02	37.66	1918	5.78	993.04	0.78	26.73	-0.72	0.38	98	55	0.03
2015	2	1	16:21:00	2015	2	1	17:43:00	01:22:00	1	72	127.95	37.93	2911	8.33	941.10	-2.43	41.15	-0.21	1.24	325	36	0.1
2015	2	4	04:40:00	2015	2	4	09:00:00	04:20:00	1	80	127.34	37.40	1948	25.71	995.53	-0.02	78.02	-0.15	1.47	184	23	0.05
2015	2	6	05:03:00	2015	2	6	07:30:00	02:27:00	0	109	129.02	35.25	8915	7.92	952.25	-4.14	69.81	-2.81	3.33	84	46	0.06
2015	2	8	13:31:00	2015	2	11	10:00:00	68:29:00	15	3809	129.17	37.17	2905	3.39	888.94	-13.06	30.86	-6.56	3.57	407	65	52
2015	2	10	10:31:00	2015	2	10	11:20:00	00:49:00	1	81	129.08	35.58	8900	6.41	908.38	-1.23	60.75	2.35	1.50	173	50	0.1
2015	2	10	15:20:00	2015	2	10	15:45:00	00:25:00	1	65	128.43	35.74	8898	32.15	948.40	5.15	43.60	3.90	0.35	38	66	0.01
2015	2	12	15:20:00	2015	2	12	16:30:00	01:10:00	2	57	126.93	37.73	1914	13.78	992.73	3.23	18.71	5.17	1.63	202	60	0.01
2015	2	12	16:35:00	2015	2	12	17:40:00	01:05:00	4	110	127.75	37.17	3893	11.07	974.82	-1.73	37.52	-0.95	4.27	59	29	0.26
2015	2	12	17:50:00	2015	2	12	19:30:00	01:40:00	1	80	127.21	37.44	1916	14.38	987.68	-2.71	31.45	-3.45	3.20	139	60	0.01
2015	2	13	12:52:00	2015	2	13	13:30:00	00:38:00	1	65	126.74	35.16	6913	17.66	995.20	2.50	46.97	10.17	1.87	48	62	0.01
2015	2	13	14:32:00	2015	2	13	16:20:00	01:48:00	0	47	126.83	37.50	1915	9.76	1010.96	3.24	28.86	2.52	1.44	42	59	0.06
2015	2	13	15:15:00	2015	2	13	16:30:00	01:15:00	2	64	126.86	35.03	6912	16.87	945.71	2.09	34.84	10.71	4.01	72	67	0.3
2015	2	13	15:20:00	2015	2	13	18:00:00	02:40:00	3	72	127.32	37.09	1948	9.65	991.18	1.39	50.81	0.33	1.03	105	60	0.01
2015	2	13	16:00:00	2015	2	13	16:45:00	00:45:00	2	96	127.86	37.08	3892	12.34	966.58	0.98	28.95	3.58	1.50	81	43	0.05
2015	2	13	16:20:00	2015	2	13	17:10:00	00:50:00	0	55	126.86	37.51	1917	9.25	972.34	0.28	36.92	-0.72	0.60	68	59	0.02
2015	2	20	14:30:00	2015	2	20	15:00:00	00:30:00	0	62	126.70	37.58	1915	9.20	1017.37	10.23	37.13	8.47	0.40	43	51	0.3
2015	2	24	12:38:00	2015	2	24	13:45:00	01:07:00	1	115	129.33	37.03	7895	5.10	938.98	5.18	38.48	21.98	1.45	58	64	0.2
2015	2	25	14:40:00	2015	2	25	15:55:00	01:15:00	1	99	127.40	36.46	4911	5.69	993.04	10.83	29.53	9.11	0.91	60	49	0.08
2015	2	25	14:59:00	2015	2	25	16:30:00	01:31:00	2	129	129.06	35.22	8915	5.35	949.68	7.56	63.02	11.20	0.13	168	64	0.01
2015	2	25	17:46:00	2015	2	25	18:25:00	00:39:00	0	113	126.76	37.47	1915	13.33	1008.87	8.20	39.83	4.87	0.90	157	57	0.06
2015	2	26	14:15:00	2015	2	26	15:35:00	01:20:00	2	97	127.96	37.39	2906	17.70	925.66	-0.88	45.71	0.00	2.13	181	56	0.2
2015	2	26	14:30:00	2015	2	26	15:05:00	00:35:00	1	48	127.04	37.86	1947	16.04	984.37	1.30	46.30	12.20	1.97	82	46	0.01
2015	3	1	12:46:00	2015	3	1	14:00:00	01:14:00	1	42	126.67	37.20	1946	23.57	1009.96	4.21	56.03	14.87	4.09	13	38	0.01
2015	3	2	14:14:00	2015	3	2	14:47:00	00:33:00	0	160	126.72	37.58	1915	7.34	1010.20	9.20	31.87	8.67	1.07	30	51	0.3
2015	3	4	11:40:00	2015	3	5	08:00:00	20:20:00	17	1420	128.75	37.52	2901	9.67	903.35	-4.64	41.97	-0.19	1.69	516	62	7
2015	3	4	15:30:00	2015	3	4	16:05:00	00:35:00	3	70	127.67	36.57	3898	11.35	944.60	-1.20	45.80	7.33	1.63	364	45	0.05
2015	3	6	12:55:00	2015	3	6	14:00:00	01:05:00	0	46	127.60	37.44	1919	18.94	1013.50	8.40	26.80	10.67	0.82	133	61	0.01

FIGURE 9. Forest-fire historical recorded events.

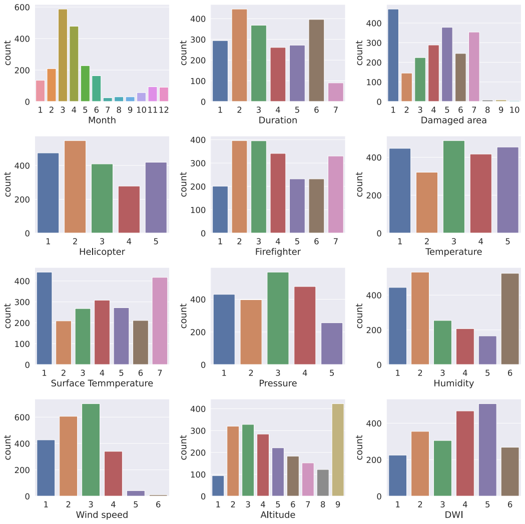


FIGURE 10. Distributions over variables in the data.

performed in accordance with the corresponding variable distribution and guideline from the forest-fire response level [2]. Figure 10 shows the number of instances for each variable. The sample distribution for the damage area class is uneven. Only a few samples are present in classes 8, 9, and 10, which correspond to significant forest-fire losses. The SMOTE [52] approach was used to oversample the minority class.

2) METRICS

The mean square error (MSE) and root mean square error (RMSE) were chosen as the loss and training metrics, respectively. The k-fold cross-validation technique was used to evaluate the model performance.

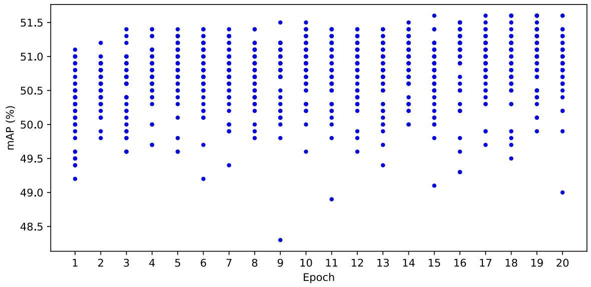


FIGURE 11. Curve of evolution search for a subnet on the trained supernet.

3) IMPLEMENTATION DETAILS

For the decision tree, the maximum depth was specified such that the tree could increase indefinitely until all leaves were pure. For the ANN, the input value was initially normalized using a batch normalization layer, followed by two hidden layers with eight hidden units; the model was then trained using an optimizer using root mean square propagation (RMSprop) with a learning rate of 2×10^{-3}. The standard BNN architecture was used in this study. The Gaussian prior with mean 0 and diagonal covariance $\theta^2 I$ on parameter θ of the network was used as follows:

$$p(\theta) = N(0, \theta^2 I) \tag{4}$$

A Gaussian prior was used to select the Gaussian approximate posterior. After the input value was normalized by using a batch normalization layer, two dense variational layers with eight hidden units were created. The training model was optimized using an RMSprop optimizer with a learning rate of 0.01.

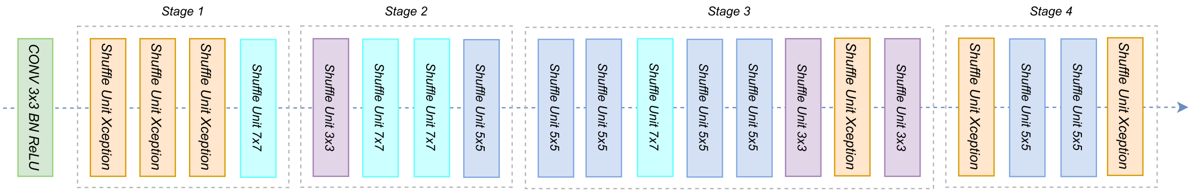


FIGURE 12. Searched architecture.

TABLE 1. Description of training variables.

Variable	Number of states	States
Month	12	Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec
Duration (minutes)	7	<=50, 50-80, 80-110,110-140,140-180, 180-480, >480
Helicopter	5	0, 1, 2, 3, >3
Firefighter	7	<=40, 40-60, 60-80, 80-100, 100-120, 120-160, >160
Temperature (^{o}C)	5	<=5, 5-10, 10-15, 15-20, >20
Surface temperature (^{o}C)	7	<=5, 5-10, 10-15, 15-20, 20-25, 25-30, >30
Pressure (hPa)	5	<=940, 940-960, 960-980, 980-1000, >1000
Humidity (%)	6	<=25, 25-35, 35-40, 40-45, 45-50, >50
Wind speed (m/s)	6	<=0.5, 0.5-1, 1-2, 2-4, 4-7, >7
Altitude (m)	9	<=30, 30-70, 70-110, 110-150, 150-190, 190-230, 230-270, 270-310, >310
DWI	6	<=30, 30-50, 50-60, 60-70, 70-80, >80
Damaged area (ha)	10	<=0.01, 0.01-0.02, 0.02-0.04, 0.04-0.08, 0.08-0.16, 0.16-0.32, 0.32-10, 10-30, 30-100, >100

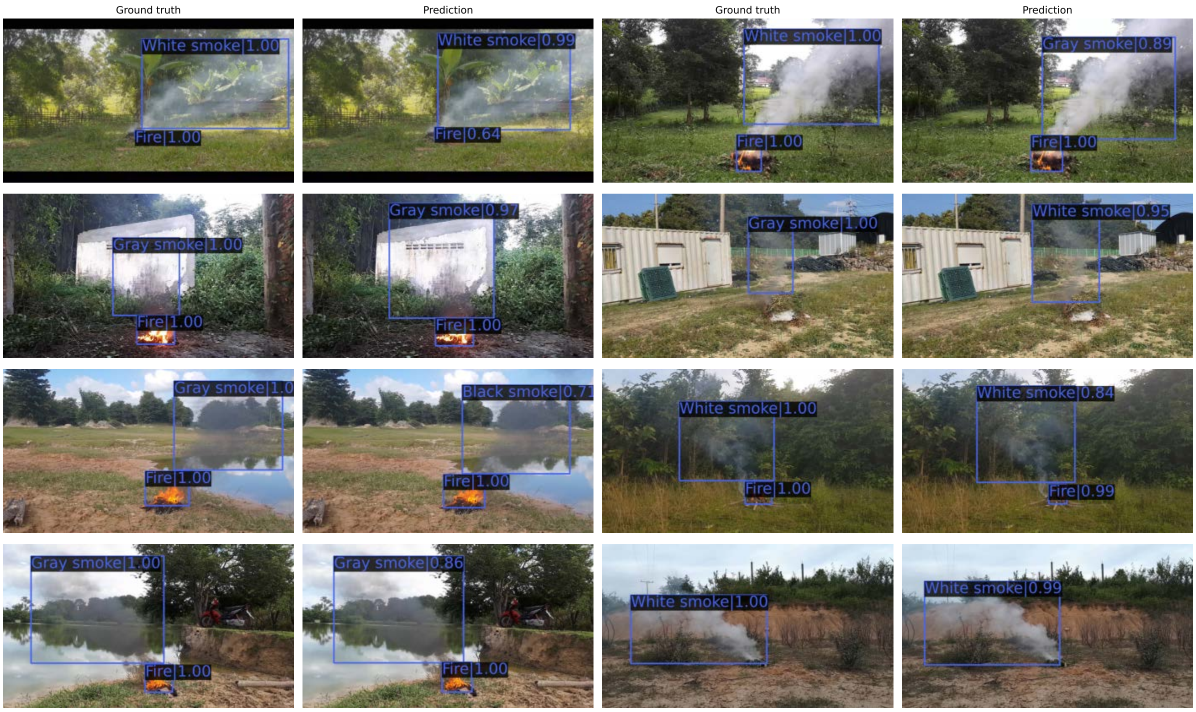


FIGURE 13. Ground truth and predicted results from Faster R-CNN with searched backbone.

V. RESULTS AND DISCUSSION

A. FOREST FIRE DETECTION

Figure 11 shows the mAP curve of the search for the subnet in the trained supernet. We used 20 epochs and 50 candidates for each epoch. The highest candidate score in each epoch increased as the evolutionary search proceeded. From this, the best architecture performance was selected. Figure 12 shows the selected architecture. The searched backbone is a combination of ShuffleBlock, which is a unit of ShuffleNetv2 [34]. The ShuffleBlock is a light-weight block based on the inference rate, not the FLOPs. Therefore, the ShuffleNetv2 shows the best performance in light-weight object detection models,

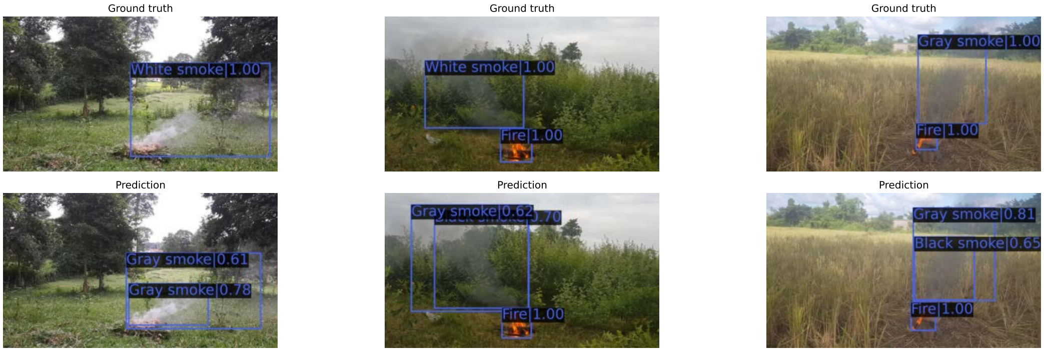


FIGURE 14. Ground truth and predicted results from Faster R-CNN with searched backbone.

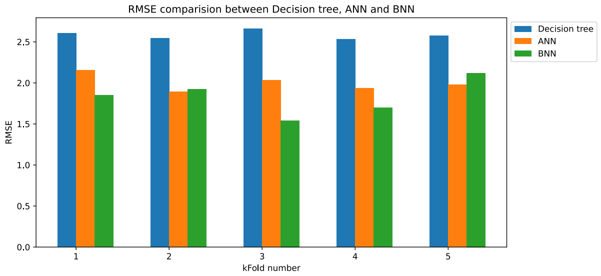


FIGURE 15. RMSE comparison between Decision Tree, ANN and BNN.

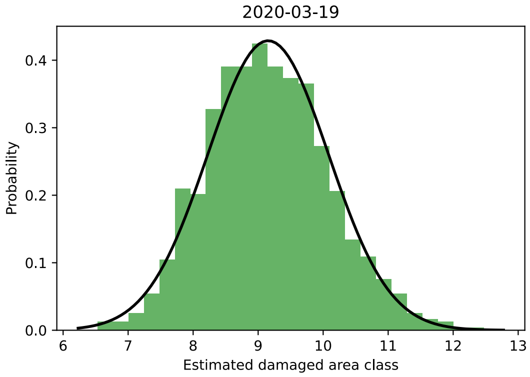


FIGURE 16. Estimated results.

such as MobileNetv2, DenseNet, Xception, NASNet. In this study, we define the search space using shuffleBlock. Each layer of our search space is a ShuffleBlock with different kernel sizes or skip layers. As shown in Figure 12, in early stage 1, the searched backbone exhibits a large kernel, and at the middle and final stages, a small kernel remains. This pattern indicates the feature extraction, the important part of object detection. As much information as possible must be extracted from input images to create a pattern useable for detection.

Table 2 lists inference results from different backbones. The test data contain nearly 40,000 forest fire images. Our searched backbone shows the highest mAP score for light-weight backbone networks. This shows that our searched backbone specialized in smoke or fire detection. That is, we avoided the limit on the accuracy of previous models using NAS.

Figure 13 shows the inference images and ground truth images from each prediction image using our searched backbone. Most prediction images are similar with the ground truth, but Figure 14 presents some exception cases.

Figure 14 shows why our model has a limited mAP score. As shown in the bounding boxes of the prediction images, in one image, the model can detect many classes, because of the visual similarities of smoke. The different prediction results in one object can lower the mAP score, which is affected by the false positive rate. However, the intention is to detect the risk factors of forest fires. From this perspective, our model can detect the risk factors well for a forest-fire response system despite its low mAP score.

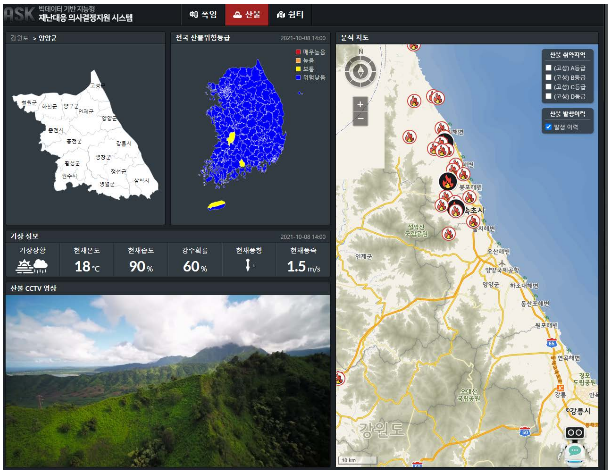


FIGURE 17. Web-based visualization.

B. DAMAGE AREA ESTIMATION USING WEATHER DATA

With the BNN, the model produces a conditional probability instead of a point-estimate prediction, from which an optimal estimate can be retrieved. Compared to other algorithms, the trained model predicts over 300 iterations and then averages the results. As illustrated in Figure 15, the decision trees with

TABLE 2. Inference results.

Backbone	Black smoke AP	Gray smoke AP	White smoke AP	Fire AP	mAP
ResNet-101-FPN [29]	22.570	5.391	19.028	46.729	23.429
ResNet-50-FPN [29]	23.920	1.463	**21.486**	46.544	23.353
VoVNetV2-99 [30]	17.848	0.248	3.215	38.923	15.058
VoVNetV2-39 [30]	15.748	1.526	12.376	40.178	17.457
VoVNetV2-19-Slim [30]	17.453	1.107	5.600	38.352	15.628
FBNetV3-G-FPN [31]	25.937	8.431	21.469	45.096	25.233
FBNetV3-A [31]	29.491	9.556	20.516	40.755	25.079
FBNetV3-A-dsmask [31]	29.495	8.784	20.229	41.820	25.082
Searched backbone	**33.900**	**9.600**	21.100	**47.100**	**27.900**

an RMSE of approximately 2.6 for each fold indicate that the model is likely to overfit the training set and provide incorrect predictions on the test set. This issue arises because of the lack of training data and inherent uncertainty in capturing forest fire history. Thus, while the ANN and BNN can achieve better results, approximately 1.7 RMSE for each fold, they still struggle with overfitting. As discussed earlier, whether the BNN has a higher accuracy or can improve performance is unclear, but it can deal with uncertainty in datasets, particularly in small datasets, similar to our approach.

To evaluate the proposed approach, a real-world forest fire occurrence - 2020 March 19 forest fire at Ulsan, South Korea - was used. According to the records, the estimated damage area of this fire was approximately 519 ha. As shown in Figure 16, the damaged region can be approximated using a distribution. This results in the following four levels of forest fire response: initial level - 0.6%, level 1 - 10%, level 2 - 34.69%, and level 3 - 54.7%. This allows for the deployment of the relevant response scenario.

VI. CONCLUSION

An appropriate forest-fire response is critical for mitigating losses and providing authorities with an effective solution. The first two stages of a forest-fire response system are early fire detection and damage area estimation. Because of the advantages of the DetNAS-based searching backbone algorithm for object detection models, the searched backbone outperform existing backbones: from hand-craft backbones, such as ResNet and light-weight VoVNet, to NAS-based FBNetV3. With an acceptable mAP of 27.9, smoke type and fire can be detected. In addition, ShuffleNetV2 blocks are considered as light-weight and effective backbones for real-time object detection. Owing to these characteristics, the searched backbone can be implemented on real-time monitoring systems. Furthermore, the damaged area can be assessed in real time using a BNN model and weather data. As illustrated in Figure 17, a web-based visualization platform was created, and weather data were updated in real time using a weather station API. When a forest fire occurs and is detected using an early fire detection model, the damaged area is approximated using the current state of the forest.

REFERENCES

[1] H. A. Wright and A. W. Bailey, *Fire Ecology: United States and Southern Canada*. Hoboken, NJ, USA: Wiley, Mar. 1982.

[2] *Korea Forest Service*, Daejeon, Republic of Korea, 2021. [Online]. Available: https://english.forest.go.kr/kfsweb/kfs/subIdx/Index.do?mn=UENG

[3] S. Kim, W. Lee, Y.-S. Park, H.-W. Lee, and Y.-T. Lee, "Forest fire monitoring system based on aerial image," in *Proc. 3rd Int. Conf. Inf. Commun. Technol. Disaster Manage. (ICT-DM)*, Dec. 2016, pp. 1–6.

[4] R. S. Priya and K. Vani, "Deep learning based forest fire classification and detection in satellite images," in *Proc. 11th Int. Conf. Adv. Comput. (ICoAC)*, Dec. 2019, pp. 61–65.

[5] D. Jung, V. Tran Tuan, D. Q. Tran, M. Park, and S. Park, "Conceptual framework of an intelligent decision support system for smart city disaster management," *Appl. Sci.*, vol. 10, no. 2, p. 666, Jan. 2020.

[6] M. Park, D. Q. Tran, D. Jung, and S. Park, "Wildfire-detection method using DenseNet and CycleGAN data augmentation-based remote camera imagery," *Remote Sens.*, vol. 12, no. 22, p. 3715, Nov. 2020.

[7] M. Park, D. Q. Tran, S. Lee, and S. Park, "Multilabel image classification with deep transfer learning for decision support on wildfire response," *Remote Sens.*, vol. 13, no. 19, p. 3985, Oct. 2021.

[8] Q.-X. Zhang, G.-H. Lin, Y.-M. Zhang, G. Xu, and J.-J. Wang, "Wildland forest fire smoke detection based on faster R-CNN using synthetic smoke images," *Proc. Eng.*, vol. 211, pp. 441–446, Jan. 2018.

[9] M. Tan, R. Pang, and Q. V. Le, "EfficientDet: Scalable and efficient object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 10781–10790.

[10] Z.-Q. Zhao, P. Zheng, S.-T. Xu, and X. Wu, "Object detection with deep learning: A review," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 11, pp. 3212–3232, Nov. 2019.

[11] J. Zhang, H. Zhu, P. Wang, and X. Ling, "ATT squeeze U-Net: A lightweight network for forest fire detection and recognition," *IEEE Access*, vol. 9, pp. 10858–10870, 2021.

[12] R. Makrigiorgis, N. Hadjittoouli, C. Kyrkou, and T. Theocharides, "AirCamRTM: Enhancing vehicle detection for efficient aerial camera-based road traffic monitoring," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2022, pp. 2119–2128.

[13] G. Plastiras, S. Siddiqui, C. Kyrkou, and T. Theocharides, "Efficient embedded deep Neural-Network-based object detection via joint quantization and tiling," in *Proc. 2nd IEEE Int. Conf. Artif. Intell. Circuits Syst. (AICAS)*, Aug. 2020, pp. 6–10.

[14] A. Kouris, C. Kyrkou, and C.-S. Bouganis, "Informed region selection for efficient UAV-based object detectors: Altitude-aware vehicle detection with CyCAR dataset," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2019, pp. 51–58.

[15] A. Moulay Akhloufi, R. B. Tokime, and H. Elassady, "Wildland fires detection and segmentation using deep learning," *Proc. SPIE*, vol. 10649, Apr. 2018, Art. no. 106490B.

[16] Y. Wang, L. Dang, and J. Ren, "Forest fire image recognition based on convolutional neural network," *J. Algorithms Comput. Technol.*, vol. 13, Jan. 2019, Art. no. 174830261988768.

[17] D. Q. Tran, M. Park, D. Jung, and S. Park, "Damage-map estimation using UAV images and deep learning algorithms for disaster management system," *Remote Sens.*, vol. 12, no. 24, p. 4169, Dec. 2020.

[18] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.

[19] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and C. A. Berg, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.* Springer, 2016, pp. 21–37.

[20] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.

[21] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[22] F. Iandola, M. Moskewicz, S. Karayev, R. Girshick, T. Darrell, and K. Keutzer, "DenseNet: Implementing efficient ConvNet descriptor pyramids," 2014, *arXiv:1404.1869*.

[23] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 2961–2969.

[24] Z. Jiao, Y. Zhang, J. Xin, L. Mu, Y. Yi, H. Liu, and D. Liu, "A deep learning based forest fire detection approach using UAV and YOLOv3," in *Proc. 1st Int. Conf. Ind. Artif. Intell. (IAI)*, Jul. 2019, pp. 1–5.

[25] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*.

[26] R. Xu, H. Lin, K. Lu, L. Cao, and Y. Liu, "A forest fire detection system based on ensemble learning," *Forests*, vol. 12, no. 2, p. 217, Feb. 2021.

[27] F. Guede-Fernández, L. Martins, R. V. de Almeida, H. Gamboa, and P. Vieira, "A deep learning based object identification system for forest fire detection," *Fire*, vol. 4, no. 4, p. 75, Oct. 2021.

[28] A. Hosseini, M. Hashemzadeh, and N. Farajzadeh, "UFS-Net: A unified flame and smoke detection method for early detection of fire in video surveillance applications using CNNs," *J. Comput. Sci.*, vol. 61, May 2022, Art. no. 101638.

[29] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[30] Y. Lee, J.-W. Hwang, S. Lee, Y. Bae, and J. Park, "An energy and GPU-computation efficient backbone network for real-time object detection," 2019, *arXiv:1904.09730*.

[31] X. Dai, A. Wan, P. Zhang, B. Wu, Z. He, Z. Wei, K. Chen, Y. Tian, M. Yu, P. Vajda, and J. E. Gonzalez, "FBNetV3: Joint architecture-recipe search using predictor pretraining," 2020, *arXiv:2006.02049*.

[32] Y. Chen, T. Yang, X. Zhang, G. Meng, X. Xiao, and J. Sun, "DetNAS: Backbone search for object detection," 2019, *arXiv:1903.10979*.

[33] T. Elsken, J. H. Metzen, and F. Hutter, "Neural architecture search: A survey," *J. Mach. Learn. Res.*, vol. 20, no. 1, pp. 1997–2017, 2019.

[34] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, "ShuffleNet V2: Practical guidelines for efficient CNN architecture design," in *Proc. Eur. Conf. Comput., Vision (ECCV)*, 2018, pp. 116–131.

[35] D. Straub, "Natural hazards risk assessment using Bayesian networks," in *Proc. 9th Int. Conf. Struct. Saf. Rel. (ICOSSAR)*, 2005, pp. 1–8.

[36] L. Blaser, M. Ohrnberger, C. Riggelsen, and F. Scherbaum, "Bayesian belief network for Tsunami warning decision support," in *Proc. Eur. Conf. Symbolic Quant. Approaches Reasoning Uncertainty*. Berlin, Germany: Springer, 2009, pp. 757–768.

[37] W. M. Dlamini, "A Bayesian belief network analysis of factors influencing wildfire occurrence in Swaziland," *Environ. Model. Softw.*, vol. 25, no. 2, pp. 199–208, Feb. 2010.

[38] Y. Dahmani and M. E.-A. Hamri, "Event triggering estimation for cell-DEVS: Wildfire spread simulation case," in *Proc. UKSim 5th Eur. Symp. Comput. Modeling Simulation*, Nov. 2011, pp. 144–149.

[39] A. L. Howes, M. Maron, and C. A. Mcalpine, "Bayesian networks and adaptive management of wildlife habitat," *Conservation Biol.*, vol. 24, no. 4, pp. 974–983, Feb. 2010.

[40] D. Hanea and B. Ale, "Risk of human fatality in building fires: A decision tool using Bayesian networks," *Fire Saf. J.*, vol. 44, no. 5, pp. 704–710, Jul. 2009.

[41] P. Papakosta, G. Xanthopoulos, and D. Straub, "Probabilistic prediction of wildfire economic losses to housing in Cyprus using Bayesian network analysis," *Int. J. Wildland Fire*, vol. 26, no. 1, p. 10, 2017.

[42] V. Sevinc, O. Kucuk, and M. Goltas, "A Bayesian network model for prediction and analysis of possible forest fire causes," *Forest Ecol. Manage.*, vol. 457, Feb. 2020, Art. no. 117723.

[43] Y. Chen, T. Yang, X. Zhang, G. Meng, X. Xiao, and J. Sun, "DetNAS: Backbone search for object detection," 2019, *arXiv:1903.10979*.

[44] B. Wu, K. Keutzer, X. Dai, P. Zhang, Y. Wang, F. Sun, Y. Wu, Y. Tian, P. Vajda, and Y. Jia, "FBNet: hardware-aware efficient ConvNet design via differentiable neural architecture search," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 10734–10742.

[45] Z. Guo, X. Zhang, H. Mu, W. Heng, Z. Liu, Y. Wei, and J. Sun, "Single path one-shot neural architecture search with uniform sampling," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 544–560.

[46] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[47] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 1321–1330.

[48] L. Valentin Jospin, W. Buntine, F. Boussaid, H. Laga, and M. Bennamoun, "Hands-on Bayesian neural networks - a tutorial for deep learning users," 2020, *arXiv:2007.06823*.

[49] C. Tymstra, B. J. Stocks, X. Cai, and M. D. Flannigan, "Wildfire management in canada: Review, challenges and opportunities," *Prog. Disaster Sci.*, vol. 5, Jan. 2020, Art. no. 100045.

[50] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2014, pp. 740–755.

[51] *OpenMMLab Model Compression Toolbox and Benchmark*, MMRazor Contributors, 2022. [Online]. Available: https://github.com/open-mmlab/mmrazor

[52] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, Jun. 2002.

DAI QUOC TRAN received the B.S. degree in transportation engineering from the Mientrung University of Civil Engineering, Vietnam, in 2018. He is currently pursuing the Ph.D. degree in civil engineering with the School of Civil, Architecture and Environmental Engineering, Sungkyunkwan University, Suwon, South Korea. His research interests include AI applications for construction safety management and natural disaster response.

MINSOO PARK received the B.S. degree in civil engineering from Sungkyunkwan University, Suwon, South Korea, in 2017, where he is currently pursuing the Ph.D. degree in civil engineering with the School of Civil, Architecture and Environmental Engineering. His research interests include AI applications and remote sensing for natural disaster response.

YUNTAE JEON is currently pursuing the double B.S. degree in civil engineering and computer engineering from Sungkyunkwan University, South Korea. His research interests include image processing and AI applications for smart city.

JINYEONG BAK received the B.S. degree in computer engineering from Sungkyunkwan University, South Korea, in 2011, and the Ph.D. degree from the School of Computing, Korea Advanced Institute of Science and Technology (KAIST), in August 2020. He has been an Assistant Professor with Sungkyunkwan University, since September 2020. His research interests include machine learning and natural language processing.

SEUNGHEE PARK received the Ph.D. degree from the Korea Advanced Institute of Science and Technology (KAIST), in 2008. He worked as a Postdoctoral Researcher at KAIST, from February 2008 to September 2008. He also worked as a Postdoctoral Research Fellow with the Center of Intelligent Material Systems and Structures, Department of Mechanical Engineering, Virginia Tech, VA, USA, from October 2008 to February 2009, where he was a Visiting Professor, from July 2009 to August 2009. He has been a Professor with Sungkyunkwan University, since March 2009.