Research and Evaluate some Deep Learning Methods to Detect Forest Fire based on Images from Camera

Cong Tung Dinh[1], Thu Huong Nguyen[1], Huyen Do Thi[2], Nam Anh Bui[3]

[1] University of Transport and Communications
[2] East Asia University of Technology
[3] High School of Education Sciences, University of Education,
Vietnam National University, Hanoi
huongnt@utc.edu.vn, tungdc@utc.edu.vn,
huyendt@eaut.edu.vn, namanhbui07@gmail.com

Abstract. Forest fires cause great consequences such as ecosystem imbalance, air quality deterioration, as well as direct impacts on human life. Early detection of a forest fire can help prevent and prevent the impact of this natural disaster and have timely remedial methods. Therefore, early forest fire detection is necessary. To accomplish this, many methods have been proposed and tested. In recent years, methods based on deep learning techniques with image data sources have been interesting and applied diversely because they can achieve optimal efficiency as well as cost savings in actual installation and operation. However, not all models give highly accurate results. In this paper, we study and evaluate four popular deep learning models (Xception, Inception-V3, VGG-19 and ResNet152-V2) that apply to forest fire detection based on images collected from cameras. With each model, we design deep learning networks to detect fires. The models were made on the dataset of 1900 images, including fire and no-fire cases. The experimental results show that all four of the above deep learning models can be applied to forest fire detection with high accuracy. In particular, the model using ResNet152-V2 gives the best results, with a fire detection capacity of 95.53%.

Keywords: Xception, Inception-V3, VGG-19, ResNet152-V2, forest fire detection.

1 Introduction

Forest fires cause serious damage, are a threat to plants, animals, and humans, and this natural disaster also has a significant impact on the environment and climate change. If the fire is not controlled and handled in time, it will cause severe consequences. Therefore, early detection of Forest fires is an urgent issue. To solve this, there have been many proposed methods. Some common methods of forest fire detection can be mentioned as using sensors (temperature sensors, smoke sensors ...) or via satellite. However, the system uses sensors with major disadvantages in small monitoring areas, and the service life of the sensors is not durable due to environmental influences and geographical conditions. For warning systems using satellites can cover a wide area, but there are some limitations such as the low resolution of images, high cost, and influenced of weather. In fact, a highly regarded method of forest fire detection is to rely on images and videos collected from CCTV cameras. Therefore, image processing techniques are studied and applied a lot in detecting fire fires.

Recently, algorithms using deep learning have been of great interest because they can detect fires quickly with high accuracy. Image recognition algorithms based on convolutional neural networks (CNN) can efficiently learn and extract complex image features. As a result, some researchers have applied CNN to fire detection through imaging. In [1], the team detected fires based on a GoogleNet model with data collected from surveillance cameras. In the paper [2], [3], [4], the researchers applied the CNN network in smoke detection. In [5], the team proposes to improve the CNN network to detect fires in real-time. The detection of fire and escape with little data was studied and proposed by A. Namozov's research team, which corrected overfitting using data augmentation techniques and generative adversarial networks [6]. In 2018, research teams compared AlexNet, VGG, Inception, ResNet, etc. models and developed smoke and flame detection algorithms [7, 9]. In 2017, Muhammad's team detected early fires using the CNN network with self-generated datasets, fine-tuned data collection cameras, and a proposed channel selection algorithm for cameras to ensure data reliability [8]. In [10] the authors studied the feasibility of network models with data collected through unmanned aerial vehicles (UAVs). Y.Luo's team [1 1] came up with a moving object detection method to create suggested zones based on background motion updates, then used the CNN algorithm to detect smoke in these areas. In [12], the authors used Gaussian (MOG) to distinguish the background and foreground, used a floor model to identify proposed smoke regions, and then applied the CaffeNet network to detect smoke. The method of using the suggested color point of the fire area, the AlexNet network to detect fire is given in the paper [13]. In 2020, the authors detected Forest fire using an LBP color signature combined with a deep learning network to detect smoke and fire with datasets collected from overhead cameras [14]. The authors in [15] offer a combined solution using LSTM and YOLO models to detect smoke in forest fire environments. The LSTM model reduces the number of layers and obtains better results in smoke detection. Another method in [16] proposes integrating cloud computing and CNN for fire detection. The paper[17] lays out a new method of classifying Forest fire based on the CNN model, called Fire_Net inspired by the AlexNet network, but improved with 15 layers, more efficient for the classification task. In [18], the authors propose a fire detection deep learning model called ForestResNet, based on ResNet-50. Another study in [19] applied a multi-layered classification model to forest fire detection. Their method uses transition learning based on VGG-16, ResNet-50, and DenseNet-121 to classify flames, smoke, fireless, and other objects in the image. The paper [20] proposes the CNN model for smoke detection in Forest fires. Their proposed CNN model applied batch normalization and multi-convolution to optimize and improve the accuracy of classification.

 ISBN: 978-604-80-8083-9

It is obviously difficult to evaluate and compare methods with each other because they are not tested on the same dataset and authors often do not publish source code details. For the purpose of comparing, evaluating as well as proposing suitable deep learning models, and fire detection applications based on images from cameras, in this paper, we study some popular deep learning models Xception, Inception-V3, VGG-19 and ResNet152-V2. Next, forest fire detection methods based on these models were installed and finally tested on the same large dataset of images collected from the camera. The test results show that these methods are all capable of good detection and the method based on ResNet152-V2 achieves the highest accuracy of over 95%.

In the next section, we present deep learning models that apply fire detection based on surveillance cameras. Part 2 describes the steps of data processing. Part 3 presents the experiment and the results achieved. Finally, part 4 is conclusive.

2 Some deep learning models that detect forest fire

2.1 Xception model

The Xception network is a deep neural network architecture introduced by researchers in the paper "Xception: Deep Learning with Depthwise Separable Convolutions" in 2016 [21]. Xception was developed from the Inception network architecture to improve and enhance performance. One of the special highlights of Xception is the use of an individual convolutional structure on each feature before performing total convolution. This helps the network learn the correlation between features. This approach reduces the number of parameters and calculations in the network, avoids overfitting, and improves model performance and accuracy. The specifics of the layers in Xception can be described as follows: The input layer receives a fixed-sized image with the parameter being the size of the input image. In the next two CONV layers, there are 32 3x3 filters and 64 3x3 filters, respectively. Next, the most important layer in Xception is the Depthwise Separable Convolution Block consisting of two stages. The first stage (Depthwise) is the individual convolution of features on each image channel, similar to traditional convolution. However, here each channel is handled independently of the others. Xceptinon then combines features from different channels to create new features (Separable). The next layer is Average Pooling with a size of 10x10 to help reduce the number of parameters and avoid overfitting. Finally, there is the fully connected layer and the output depends on the number of layers of the classification problem. For the dataset in the paper, Xception predicts up to 94.26% accuracy.

2.2 Inception-V3 model

The Inception-V3 model is a deep neural network architecture developed and published in 2015 [22]. InceptionV3 is an improved version of earlier versions such as Inception and Inception-V2. The special feature of Inception-V3 is the use of the Inception module, a module with many parallel branches that allows the model to become flexible and can learn the features of the image at various scales. The Inception-V3 model also uses common techniques such as regularization, dropout, and batch normalization to reduce overfitting during model training. Inception-V3 has 48 convolutional layers and 3 fully connected layers as follows: The input layer that receives the input data is an image, and the pixel values are normalized to range from -1 to 1. This is followed by 48 convolutional layers, with kernels of size 3x3 or 5x5, stride=2. In the max pooling layer, the model applies with kernel 3x3, stride=2 to reduce the input size and retain important features of the image. The peculiarity of the model lies in the Inception Module layer with many parallel branches. Each Inception Module has 4 branches including The 1x1 Convolution branch uses a 1x1 size kernel used to reduce the input size before applying other convolution layers. 3x3 Convolution Branches with Different Padding: These 3x3 convolution layers are applied with the same or valid padding to ensure the output size of these branches is equal. The branch uses a 5x5 kernel to learn features at a greater scale than 3x3 convolutional layers. Branch Max Pooling with 1x1 Convolution, kernel size 3x3 to reduce the dimension of the output. The outputs of these branches are then joined together in-depth using concatenation. After the adoption of several Inception layers, an Average Pooling layer is used to reduce the size of the output and retain important features. Finally, there are the Fully Connected, Softmax, and Dropout layers. For the dataset in the paper, the prediction InceptionV3 model has an accuracy of 94.21%.

2.3 VGG-19 model

The VGG-19 model is a convolutional neural network (CNN) developed by the Visual Geometry Group (VGG) research group at the University of Oxford. VGG-19 is an upgraded version of the VGG-16 model, with the main difference being the number of layers. The architecture of VGG-19 is built from convolution layers and pooling layers, including 16 convolutional layers and 3 fully connected layers. The convolutional layers are divided into five groups, where each group has layers of the same input and output dimensions. Before each group of convolutional layers, there is a max pooling layer to reduce the size of the input. The convolutional layers in VGG-19 use the Relu activation function, with filters of small size of 3x3 with a stride is 1 and padding is 1. After the convolution and max pooling layers are done, the features are extracted from the image and put into fully connected layers with the neurons in the network. Finally, the final layer uses the softmax function to classify the images into different labels. For the dataset in the paper, the prediction VGG-19 model has an accuracy of 94.73%.

2.4 ResNet152-V2 model

The ResNet152-V2 model was developed by the Microsoft Research team and published in 2017 [23]. ResNet152-V2 is an improved version of the ResNet152 model, improving the learning of the model and reducing computational complexity. The ResNet152-V2 model consists of 152 layers of neural networks, including convolutional layers, activation layers, pooling layers, and full connection layers.

 ISBN: 978-604-80-8083-9

Each convolutional layer of the model is applied several filters to search for features in the input image. The ResNet152-V2 model uses the Residual block proposed by ResNet, which avoids the deterioration of accuracy when the neural network becomes very deep. The ResNet152-V2 model also uses a bottleneck technique, allowing the model to learn higher-resolution features without adding depth to the network. This technique uses small-sized convolutional layers before using large-sized convolutional layers to reduce calculation costs. The specifics of the models are described as follows: The input layer is a 224x224x3 image. The convolutional layer consists of 152 layers divided into different blocks. Each block consists of multiple convolutional layers and activation layers. The Residual blocks of the ResNet152-V2 Residual network contain a number of building blocks designed to reduce the depth of the network and reduce the deterioration of accuracy when the network becomes very deep. Finally, there are trigger functions, pooling layers, and full connection layers. ResNet152-V2 is an upgraded version of ResNet152, with improvements in architecture and training methods to achieve greater accuracy. It has 152 layers and 8.0 million parameters, trained on an ImageNet dataset of 1.28 million images. For the dataset in the paper, the prediction ResNet152-V2 model has an accuracy of 95.53%

3 Dataset

In this paper, we use Ali Khan's forest fire dataset, with three color channels, 250x250 in size, for a total of 1900 images, divided into two layers, 950 fire images and 950 no-fire images [24]. Figure 1 depicts some of the images in the dataset. The data is preprocessed and labeled, divided into two parts with 80% of the samples used for training and 20% of the samples used for testing. In the test data set, we use 20% for validation. The allocation of training and testing data is presented in Table 1. Training data will be trained using different deep learning algorithms, thereby making comparisons and judgments about the methods used.

Fig 1. Some images in the dataset (Row 1 is some images of the fire class, row 2 is some images of the no-fire class.)

Table 1. Distribution of data in the article

Dataset	Fire	No-fire	Total
Training	608	608	1216
Test	190	190	380
Validation	152	152	304
Total	950	950	1900

The training process of all four models is similar and is played out as follows: the image is preprocessed by turning into a grayscale image to reduce the number of dimensions of the input matrix and convert the image to a common size of 224x224. In addition, to better represent the diversity of images, we perform training dataset augmentation using Keras' "ImageDataGenerator". The images are cloned, perform a 20-degree rotation, scaling, shear transformation, translation, zoom 20%, flip, and then put into a deep learning model for training. Figure 2 depicts several images after data augmentation. At the end of the training, we will have a model aimed at detecting forest fires.

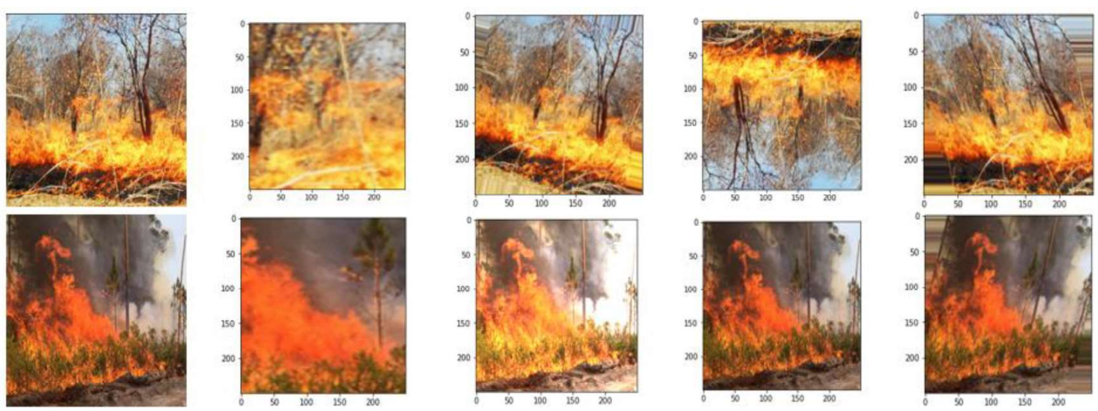

Fig 2. Some images after data augmentation

4 Results and evaluation

In this section, we present the empirical results of deep learning methods applied to forest fire detection. Our models are built on computers with CORE I7-10700 2.9GHZ configuration, 16 GB RAM, Windows 10 OS, Python 3.6, and TensorFlow with a Learning Rate of 1e-4, batch_size=32, epochs = 100.

After testing four deep learning models, we found the ResNet152-V2 based method to be the most accurate (95.53%). Besides, the methods using VGG-19 have an accuracy of 94.73%, Inceptionp-V3 has an accuracy of 94.21% and Xception has an accuracy of 93.94%. To compare and evaluate four deep learning models, we performed statistics and compared four values after training the models including Precision, Recall, F1 Score, and Accuracy. In particular, Accuracy is the ratio of the number of correctly predicted data points to the total number of data points in the test set. Precision is the ratio of the number of points correctly identified in a class to the

 ISBN: 978-604-80-8083-9

total number of points classified in that class. The Recall value is the ratio of the number of correctly identified points in a class to the total number of data points in that class. The F1-score quantity is the harmonic average determined based on two measurements. Results comparing fire detection accuracy between algorithms are shown in Table 2.

Table 2. Comparison of accuracy between algorithms

Models	**Accuracy**	**Precision**	**Recall**	**F1-Score**
ResNet152-V2	95.53%	92.61%	98.95%	95.67%
VGG-19	94.73%	94.32%	96.31%	95.31%
Inception-V3	94.21%	91.59%	97.37%	94.39%
Xception	93.94%	92.82%	95.26%	94.03%

To better clarify the results of each deep learning technique, we use the Confusion matrix that displays the classification performance of each deep learning model more visually. Accordingly, the matrix's vertical axis corresponds to the two layers of fire and no-fire. Diaphragmatic axes are labels according to the prediction model, which also corresponds to the two classes above.

In the ResNet152-V2 model, the Confusion matrix is shown in Figure 3. Accordingly, the model correctly predicted 188 fire cases out of 203 predicted cases. There were 15 cases where the model wrongly predicted fire despite no-fire. There are 2 cases of forest fires, but the model is not predictable. Out of a total of 190 real fire cases, the model correctly predicted 188 cases.

In the VGG-19 model, the Confusion matrix is shown in Figure 4. Accordingly, the model correctly predicted 183 fire cases out of 194 predicted cases. There were 11 cases where the model wrongly predicted fire despite no-fire. There are 7 cases of forest fires, but the model is not predictable. Out of a total of 190 real fire cases, the model correctly predicted 183 cases.

In the InceptionV3 model, the Confusion matrix is shown in Figure 5. Accordingly, the model correctly predicted 185 fire cases out of 202 predicted cases. There were 17 cases where the model wrongly predicted fire despite no-fire. There are 5 cases of forest fires, but the model is not predictable. Out of a total of 190 real fire cases, the model correctly predicted 185 cases.

In the Xception model, the Confusion matrix is shown in Figure 6. Accordingly, the model correctly predicted 181 fire cases out of 195 predicted cases. There were 14 cases where the model wrongly predicted fire despite no-fire. There were 9 cases of forest fires, but the model was not predictable. Out of a total of 190 real fire cases, the model correctly predicted 181 cases.

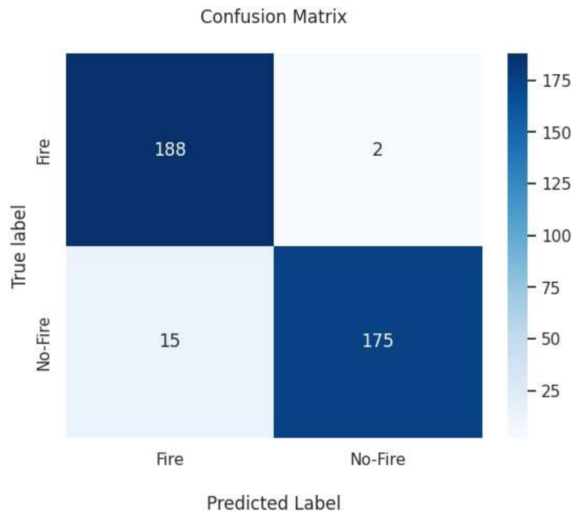

Fig 3. Confusion matrix of ResNet152-V2 model

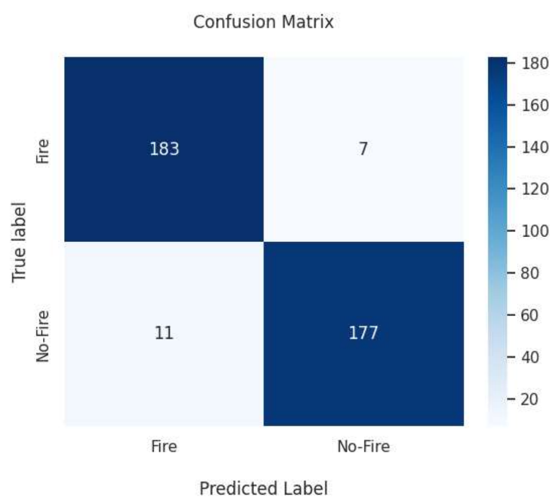

Fig 4. Confusion matrix of VGG-19 model

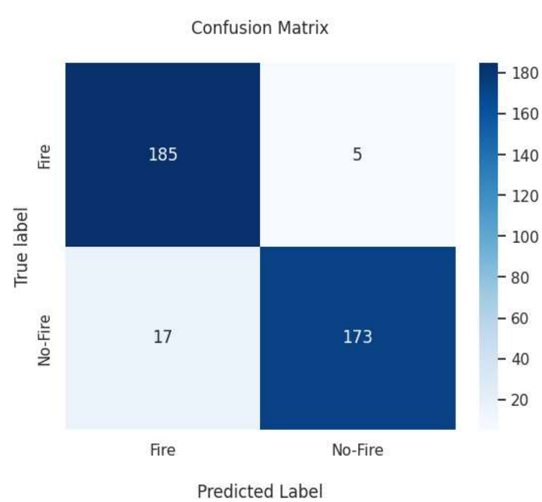

Fig 5. Confusion matrix of Inception-V3 model

 ISBN: 978-604-80-8083-9

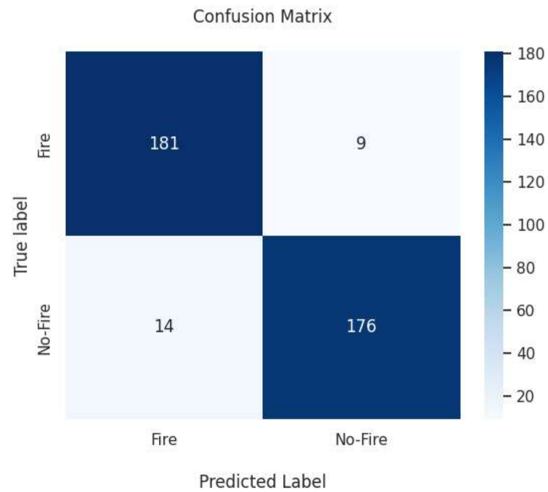

Fig 6. Confusion matrix of Xception model

As can be seen, all four models are highly accurate in applying forest fire detection based on images collected from the camera. The ResNet152-V2 model is the most efficient, with little difference in Precision and Recall ratios, although it incorrectly predicted 15 cases, ResNet152-V2 only missed 2 cases out of 190. This suggests that ResNet152-V2 is the model that matches the dataset used in the paper. In the VGG-19 model, the Precision value is the highest, demonstrating the model's effectiveness in fire detection for predictive data samples. However, if based on total actual data, the Recall value is lower than the ResNet-152 model. In the Xception model, it may be because the depth convolution layer is not effective, when the dataset with large fire images, the surrounding background has the same color and features as fire, creating confusion, leading to the combination of features on each image channel of Xception is not highly effective. Therefore, in the Confusion matrix, the number of cases mistakenly predicted to be fire and not fire is not much different (14 cases and 9 cases). In the Inception-V3 model, although the accuracy is slightly higher than that of the Xception, the difference between Precision and Recall is large, resulting in an F1-Score measurement almost on par with the Xception model.

5 Conclusions

Early detection of forest fires is an urgent need today. The paper presented and evaluated four models of applying deep learning techniques ResNet152-V2, VGG-19, Inception, and Xception applying forest fire detection through images collected from cameras. The test results show that the above methods are highly accurate, in which, the forest fire detection rate of the Xception model is 93.94%, the Inception-V3 model is 94.21%, the VGG-19 model is 94.73% and the ResNet152V2 model is 95.53%. The results also show that the fire detection program with the ResNet152-V2 model has the highest accuracy. In the future, we will develop and improve the deep learning network model to achieve greater accuracy, which can be effectively applied in forest fire detection and warning.

Acknowledgements

This research is funded by University of Transport and Communications (UTC) under grant number T2023-CN-004.

References

1. K. Muhammad, J. Ahmad, I. Mehmood, *et al.* Convolutional neural networks based fire detection in surveillance videos, IEEE Access, 6 (2018), pp. 18174-18183.
2. C. Tao, J. Zhang, P. Wang Smoke detection based on deep convolutional neural networks, 2016 International Conference on Industrial Informatics - Computing Technology, Intelligent Technology, Industrial Information Integration (ICIICII) (2016), pp. 150-153.
3. A. Filonenko, L. Kurnianggoro, K. Jo Comparative study of modern convolutional neural networks for smoke detection on image data 2017 10th International Conference on Human System Interactions (HSI) (2017), pp. 64-68.
4. Z. Yin, B. Wan, F. Yuan, *et al.* A deep normalization and convolutional neural network for image smoke detection IEEE ACCESS, 5 (2017), pp. 18429-18438.
5. A.J. Dunnings, T.P. Breckon, Experimentally defined convolutional neural network architecture variants for non-temporal real-time fire detection, 2018 25th IEEE International Conference on Image Processing (ICIP) (2018), pp. 1558-1562.
6. A. Namozov, Y. Cho, An efficient deep learning algorithm for fire and smoke detection with limited data, Adv. Electr. Comput. Eng., 18 (2018), pp. 121-128.
7. W. Mao, W. Wang, Z. Dou, Y. Li, Fire recognition based on multi-channel convolutional neural network, Fire Technol., 54 (2018), pp. 531-554.
8. K. Muhammad, J. Ahmad, S.W. Baik, Early fire detection using convolutional neural networks during surveillance for effective disaster management, Neurocomputing, 288 (2018), pp. 30-42.
9. Y. Hu, X. Lu, Real-time video fire smoke detection by utilizing spatial-temporal ConvNet features Multimed. Tool. Appl., 77 (2018), pp. 29283-29301.
10. L. Wonjae, K. Seonghyun, L. Yong-Tae, L. Hyun-Woo, C. Min, Deep neural networks for wild fire detection with unmanned aerial vehicle, 2017 IEEE International Conference on Consumer Electronics (ICCE) (2017), pp. 252-253.
11. Y. Luo, L. Zhao, P. Liu, D. Huang, Fire smoke detection algorithm based on motion feature and convolutional neural networks, Multimed. Tool. Appl., 77 (2018), pp. 15075-15092.
12. N.M. Dung, D. Kim, S. Ro, A video smoke detection algorithm based on cascade classification and deep learning, KSII Internet Inf., 12 (2018), pp. 6018-6033.
13. Z. Zhong, M. Wang, Y. Shi, W. Gao, A convolutional neural network-based flame detection method in video sequence, Signal Image Video Process., 12 (2018), pp. 1619-1627.
14. Hossain, F.M.A.; Zhang, Y.M.; Tonima, M.A. Forest fire flame and smoke detection from UAV-captured images using fire-specific color features and multi-color space local binary pattern. J. Unmanned Veh. Syst. 2020, 8, 285–309.
15. Jeong, M.; Park, M.; Nam, J.; Ko, B.C. Light-Weight Student LSTM for Real-Time Forest fire Smoke Detection. Sensors 2020, 20, 5508.

 ISBN: 978-604-80-8083-9

16. Srinivas, K.; Dua, M. Fog Computing and Deep CNN Based Efficient Approach to Early Forest Fire Detection with Unmanned Aerial Vehicles. In Proceedings of the International Conference on Inventive Computation Technologies (ICICIT), Tamil Nadu, India, 29–30 August 2019; Smys, S., Bestak, R., Rocha, A., Eds.; Springer Nature: Chennai, India, 2019; pp. 646–652.
17. Zhao, Y.; Ma, J.; Li, X.; Zhang, J. Saliency Detection and Deep Learning-Based Forest fire Identification in UAV Imagery. Sensors 2018, 18, 712.
18. Tang, Y.; Feng, H.; Chen, J.; Chen, Y. ForestResNet: A Deep Learning Algorithm for Forest Image Classification. J. Physics Conf. Ser. 2021, 2024, 012053.
19. Park, M.; Tran, D.Q.; Lee, S.; Park, S. Multilabel Image Classification with Deep Transfer Learning for Decision Support on Forest fire Response. Remote Sens. 2021, 13, 3985.
20. Sun, X.; Sun, L.; Huang, Y. Forest fire smoke recognition based on convolutional neural network. J. For. Res. 2020, 32, 1921–1927.
21. Chollet, F. Xception: Deep Learning with Depthwise Separable Convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; IEEE: Piscataway, NJ, USA; pp. 1800–1807.
22. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the Inception Architecture for Computer Vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; IEEE: Piscataway, NJ, USA, 2016, pp. 2818–2826.
23. He, K.; Zhang, X.; Ren, S.; Sun, J. Identity Mappings in Deep Residual Networks. In Proceedings of the European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 11–14 October 2016, Springer International Publishing: New York, NY, USA; pp. 630–645.
24. A. Khan, B. Hassan, S. Khan, R. Ahmed and A. Adnan, "DeepFire: A Novel Dataset and Deep Transfer Learning Benchmark for Forest Fire Detection," Mobile Information System, vol. 2022, pp. 5358359, 2022.