Wildfire Detection Via Transfer Learning: A Survey

Ziliang Hong (zhong37@uic.edu)
University of Illinois at Chicago
Emadeldeen Hamdan
University of Illinois at Chicago
Yifei Zhao
University of Illinois at Chicago
Tianxiao Ye
University of Illinois at Chicago
Hongyi Pan
University of Illinois at Chicago
Ahmet Enis Cetin
University of Illinois at Chicago

Research Article

Keywords: Wildfire Detection, Transfer Learning, Convolutional Neural Network, Vision Transformer

Posted Date: July 7th, 2023

DOI: https://doi.org/10.21203/rs.3.rs-3137242/v1

Additional Declarations: No competing interests reported.

Version of Record: A version of this preprint was published at Signal, Image and Video Processing on August 30th, 2023. See the published version at https://doi.org/10.1007/s11760-023-02728-3.

Wildfire Detection Via Transfer Learning: A Survey

Ziliang Hong, Emadeldeen Hamdan, Yifei Zhao, Tianxiao Ye, Hongyi Pan, Ahmet Enis Cetin

Department of Electrical and Computer Engineering, University of Illinois Chicago, Harrison, Chicago, 60607, Illinois, United States.

Contributing authors: zhong37@uic.edu; ehamda3@uic.edu; yzhao248@uic.edu; tye24@uic.edu; hpan21@uic.edu; aecyy@uic.edu;

Abstract

This paper presents a comprehensive survey of publicly available neural network models specifically designed for detecting wildfires using regular visible-range cameras positioned on hilltops or forest lookout towers. The surveyed models are first pre-trained on the ImageNet-1K dataset and then fine-tuned on a custom wildfire dataset to enhance their performance. Evaluations are conducted on diverse wildfire images, enabling a thorough assessment of their capabilities. The survey findings provide valuable insights for individuals interested in leveraging transfer learning techniques for wildfire detection. Among the examined models, Swin Transformer-tiny achieves the highest Area Under the Curve (AUC) value, indicating strong overall performance in distinguishing wildfire events. However, ConvNext-tiny stands out for its exceptional ability to detect all instances of wildfires while maintaining the lowest false alarm rate within our dataset. These results highlight the varying strengths of different neural network models and offer valuable guidance for selecting an appropriate model based on specific detection requirements and priorities.

Keywords: Wildfire Detection, Transfer Learning, Convolutional Neural Network, Vision Transformer

1 Introduction

Early detection of wildfires is crucial in minimizing the harm they cause to people and the economy. Researchers have developed different techniques, including real-time algorithms that use video-based surveillance systems and deep neural networks that can recognize fire and smoke images[1–13]. Some of these methods enable a single camera to detect wildfire smoke in real-time from a distance[14].

Neural network-based methods for wildfire detection eliminate the need for manual feature selection, but they require a lot of data and computing power. To mitigate the issue of insufficient data, synthetic data is used for training, and transfer learning techniques like Fast R-CNN and Yolo series algorithms[15] are utilized to enhance the model's performance and reduce the amount of data required[16, 17]. Transfer learning has been used in this survey, inspired by previous papers[14, 18], leading to the development of effective neural network models for forest fire detection since 2015.

In this paper, we explore the feasibility of several classical models in the field of wildfire detection and compare their performance. These models are Residual Neural Network V2(ResNetV2), Data-efficient, image Transformers (DeiT), EffecientNetV2, Big Transfer (BiT), MobileNetV3, Swin Transformer, and ConvNeXt[19–26]. Inspired by a previous paper [14], a single image is split into sub-images for object detection. We evaluate critical indicators, including accuracy, false alarm rate, true detection rate, detection latency, and implementation latency for wildfire

detection [14, 18]. Experiments compare model performance, analyze superior models, and provide a summary of comparisons.

2 Overview of Artificial Neural Networks

In this section, we present a comprehensive review of the selected models, detailing their respective structures and innovations.

2.1 Residual Neural Network (ResNet)

ResNet is a revolutionary neural network [27]. It achieved unprecedented depth with more than 1000 layers, owing to the introduction of residual blocks [27]. Before the ResNet, it is always challenging to design very deep Convolutional Neural Networks (CNNs) because neural networks that are too deep usually face the problems of gradient vanishing and exploding. Besides, an increasing number of layers could also cause degeneration. Through the skip connection structure of layers which is called residual block and the usage of the Batch Normalization (BN) layer, ResNet shows its ability to solve this predicament. The core equation of a residual block in ResNet can be expressed as follows:

$$y = \mathcal{F}(x, \{W_i\}) + W_s x. \tag{1}$$

In ResNetv2 [19], the structure of the residual module was improved. The BN layer and activation function are placed in front of the weight layer as preactivation. This is the main difference between ResNetv2 and ResNetV1. Such a structure can not only afford efficient backpropagation but also allow the BN layer to play a regularization role, which makes ResNetV2 significant progress.

2.2 MobileNet

MobileNet [28] is built primarily from depthwise separable convolutions. Depthwise separable convolutions include a standard convolution into a depthwise convolution and a 1×1 convolution called a pointwise convolution. It can reduce computation and model size effectively.

Meanwhile, the network structure has been optimized and a parameter called width multiplier has been introduced to further thin the network uniformly at each layer. MobileNet is small and has low latency. It is implementable on computationally limited platforms and shows strong performance. The authors introduce MobileNetV2 using the inverted residual with the linear bottleneck to further improve the network [29]. In the next generation MobileNetV3, a combination of these modules (depthwise separable convolutions and inverted residual with linear bottleneck) are used in the first two generations. Platform-aware Neural Architecture Search (NAS) is also employed to search for the global network structures and then use the NetAdapt algorithm to search per layer for the number of filters. Computationally expensive layers has been redesigned and a nonlinearity called swish to replace ReLU is used instead. As the next generation of MobileNet, MobileNetV3 can achieve higher accuracy and lower latency than MobileNetV2 [24].

2.3 Big Transfer (BiT)

The innovation of BiT lies in its large-scale pretraining strategy that involves training on multiple public datasets. By leveraging this strategy, BiT achieves high performance on a wide range of computer vision tasks and surpasses previous state-of-the-art results. Also, Group Normalization and weight standardization is used instead of Batch Normalization. Because it incurs inter-device synchronization costs when using distributed training. And it is detrimental to transfer due to the requirement to update running statistics.

During the transfer to downstream tasks, a fine-tuning protocol called BiT-HyperRule is proposed. It is heuristic to set the following hyperparameters per-task: training schedule length, resolution, and whether to use MixUp regularization. The models are evaluated on standard benchmarks and have a good performance. The recipe is simple and effective when we transfer pre-trained models to diverse tasks [23].

2.4 EffecientNet

The compound scaling method is proposed to scale network width, depth, and resolution with a set of fixed scaling coefficients. NAS is used to design a new baseline network and scale it up to obtain a family of models called EfficientNets, which is smaller and faster than existing convolutional neural networks [30].

EfficientNetV2 is introduced in June 2021. A combination of training-aware NAS and scaling are used to improve both training speed and parameter efficiency. They design a search space enriched with additional ops such as Fused-MBConv and propose an improved method of progressive learning, which can adjust regularization along with image size. EfficientNetV2 have up to 11x faster training speed and up to 6.8x better parameter efficiency on ImageNet, CIFAR, Cars, and Flowers dataset, than prior art such as ResNet-101 and ViT-L/16 (21k) [22].

2.5 Data-efficient image Transformers (DeiT)

Recurrent Neural Networks (RNNs) require information from previous or next-time steps for calculations, making parallel computation difficult and limiting them to serial processing. By adopting the Self-Attention mechanism [31], The Transformer model for sequence processing avoids horizontal propagation, relying instead on vertically stacked self-attention layers that allow for parallel computation and acceleration using GPUs.

Vision Transformer (Vit) is a transformer-based model in computer vision which require massive training data [32]. In order to overcome the limitations of Vit, the Data-efficient image transformers (DeiT) model was developed. A systematic optimization and regularization approach was employed on DeiT, which includes data augmentation during training. The authors employed the so-called soft and hard-label knowledge distillation to facilitate the teacher model in guiding DeiT during training [20, 21].

2.6 Swin Transformer

Swin Transformer uses a hierarchical construction method similar to CNNs. Such a backbone helps to build detection and segmentation tasks on this basis[25]. When using the Windows Multi-head Self-Attention (W-MSA) module, the self-attention calculation will only be performed within each window, so there is no information transfer from window to window. To solve this problem, the authors introduced the Shifted Windows Multi-Head Self-Attention (SW-MSA) module. Relative Position Bias is also employed to improve the performance of the model.

2.7 ConvNeXt

ConvNeXt does not introduce novel architectural or methodological innovations. Instead, it leverages existing techniques and optimized CNNs for enhanced performance. The authors first used the strategy of training ViT to train the original ResNetV2-50 model and observed a significant improvement in performance compared to the baseline. This benchmark performance was then utilized for subsequent experiments. Through a series of experiments, ConvNeXt has faster inference speed and higher accuracy compared with Swin Transformer with the same computational complexity [26].

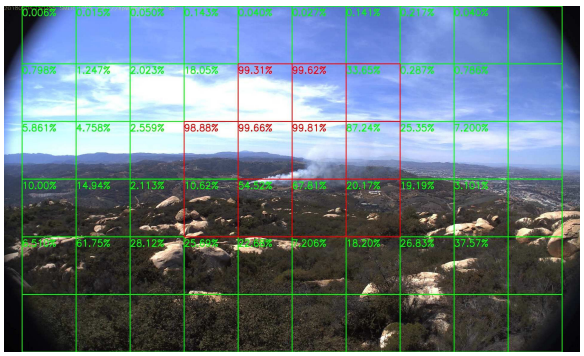

Figure 1: The example of detection result (detected by Swin Transformer-tiny).

3 Methodology

3.1 Methods of Implementation

Object detection tasks involve identifying the location and classification of objects in images. Usually, the bounding box-based method needs to manually label massive bounding boxes like the Yolo series algorithm[15]. In this experiment, the approach taken is to divide each image into 45 blocks as Fig. 1. Using this approach, it is possible to detect and locate fires accurately and we do not need to label the bounding box of the wildfire. Assuming that the dimensions of the active image area are M_i and N_i and that the dimensions of each block are M_b and N_b, an equation can be used to express the row number R and column number C for each block:

$$R = \lfloor \frac{M_i}{M_b} \rfloor, C = \lfloor \frac{N_i}{N_b} \rfloor, \tag{2}$$

where $\lfloor \cdot \rfloor$ stands for the floor function. The main task is to construct a binary classifier that can predict the whether there is a wildfire or not. Forest fire detection devices are often used in remote wilderness areas where weight and computational resources are limited. Therefore, small models are chosen to fit these devices.

3.2 Dataset

In this work, we build a wildfire dataset by enriching the dataset in [14] and [18] to about 35k images. The training subset approximately contains 14k images of normal forests and 9k wildfire images. The test subset contains about 8k images of normal forests and 4k wildfire images. All the data images are from the HPWREN wildfire dataset , the FIRESENSE database [33], google images, and YouTube videos. Furthermore, to evaluate the performance in real-world applications, we evaluate the models on the HPWREN videos to estimate the detection latency. Fig. 2 and Fig. 3 show some samples from the dataset.

(a) (b) (c) (d)

Figure 2: Examples of the dataset images. Wildfire exists in (a) and (b). Wildfire does not exist in (c) and (d).

Figure 3: HPWREN samples for test model detection. We use data from 9 HPWREN cameras, each of which recorded the process of occurrence of fire.

4 Experiment

4.1 Training Models

Transfer learning techniques in deep learning can address the time-consuming task of gathering extensive training data and overfitting problems In this study, pre-trained models from the ImageNet-1K open-source database [34] were used for forest fire detection using the TensorFlow deep learning library with an NVIDIA RTX3070 GPU. The training process consisted of 15 epochs with a fixed feature extractor method used in the first 10 epochs and a transfer learning strategy applied in the final 5 epochs. Fine-tuning all the layers in the whole model further improved the models' performance while reducing the demand for large amounts of data. After fine-tuning, the probability of wildfire was calculated by feeding the results to a softmax layer. The study assumed that images with fire are negative samples and images without fire are positive samples. To ensure practical application, the wildfire detection model should not be overly sensitive while accurately detecting forest fires to the greatest extent possible using the true detection rate and false alarm rate in the confusion matrix.

4.2 Testing Models

4.2.1 Evaluation Indicators

The evaluation of deep neural networks (DNNs) involves five key indicators: accuracy, true detection rate, false alarm rate, floating point operations (FLOPs), number of parameters, detection latency, and Receiver Operating Characteristic (ROC). ROC curve is a graph that shows how sensitive a model is to various threshold ranges between 0% and 100% [35]. By calculating the False Positive Rate (FPR) and True Positive Rate (TPR) under different thresholds and setting them as X-axis and Y-axis respectively, ROC curve and Area Under the Curve (AUC) can be used to compare the performance of different models and usually a larger AUC indicates a better performance. The mathematical expressions for ROC and AUC can be represented as:

$$\mathrm{ROC} = \frac{\mathrm{TPR}}{\mathrm{TPR} + \mathrm{FPR}} \tag{3}$$

$$\mathrm{AUC} = \int_0^1 \mathrm{ROC}(x)\,\mathrm{d}x \tag{4}$$

Accuracy measures the percentage of accurate predictions generated by the neural network, indicating the models' fitness for purpose. True detection rate and false alarm rate are crucial indicators in wildfire detection systems, with true detection rate representing the proportion of correct negative predictions made by the model and false alarm rate indicating the percentage of false alarms, with a lower false alarm rate indicating greater reliability.

Parameters and FLOPs are essential factors affecting neural networks, with the former indicating the model's complexity and the latter measuring the number of calculations an algorithm can perform in a second. FLOPs are also used to estimate the computational requirements of training or executing a neural network on a hardware device.

Finally, detection latency refers to the interval between the start of an actual fire and the system's detection of it. In experiments, this metric is evaluated in frames rather than seconds. These indicators can assist in evaluating the potential applicability and performance of DNNs, which can inform the development of more reliable and efficient models.

4.2.2 Basic Test

Fig. 4 shows the ROC curve of each model. Because the curves are dense, Fig. 4 displays the details of

Model	Parameters (M)	FLOPs (G)	AUC	ACC	True Detection	False Alarm
Swin Transformer-tiny	27.6	8.99	**0.99917**	97.95%	94.63%	0.36%
DeiT-tiny	5.5	2.54	0.99876	**98.13**%	95.12%	0.35%
ConvNeXt-tiny	27.8	8.99	0.99743	96.69%	90.52%	0.18%
MobileNetV3-small	20.3	**0.12**	0.99114	95.17%	94.11%	4.30%
BiT-small	23.5	8.38	0.98973	96.86%	**99.78**%	4.63%
ResNetV2-50	23.6	6.99	0.98837	95.00%	98.92%	6.99%
EfficientNetV2	**1.5**	5.74	0.98571	93.47%	96.48%	**8.06**%
Mobilenet-edgetpu-v2	2.5	1.03	0.98056	94.70%	91.73%	3.79%

Table 1: Parameters, FLOPs, AUC of models. Accuracy, True Detection Rate, and False Alarm Rate of models on the test dataset. Models are arranged in descending order of AUC.

Video No.	1	2	3	4	5	6	7	8	9	Average
Swin Transformer-tiny	7	4	**0**	27	6	**1**	2	4	2	5.8
DeiT-tiny	5	4	1	29	7	2	6	4	2	6.7
ConvNeXt-tiny	5	4	**0**	29	7	**1**	6	6	**1**	6.6
MobileNetV3-small	6	11	**0**	35	7	4	27	13	**1**	11.6
BiT-small	**2**	**0**	**0**	**26**	**4**	**1**	**1**	3	**1**	**4.2**
ResNetV2-50	3	2	**0**	29	5	**1**	8	7	2	6.3
EfficientNetV2	4	2	**0**	26	6	4	6	**2**	**1**	5.7
Mobilenet-edgetpu-v2	6	11	1	35	7	4	6	11	**1**	9.1

Table 2: Considering the rigor of fire prediction, we set the threshold to 95%. Each number represents the frame number at which the models first detected the presence of fire.We Calculate detection latency as the time between the frame when the fire is first detected and the frame when the fire starts. In the last column, we calculate the average detection latency.

the top left part. Table 1 shows the FLOPs and parameters of each model, the models are arranged in the order of AUC size, from largest to smallest. Table 1 records the performance of each model on the test dataset with the threshold of 95%. As Table 1 shows, DeiT-tiny owns the highest accuracy which is 98.13% followed by 97.95% from Swin Transformer-tiny and they are both transformer-based models. The remaining traditional CNNs models perform slightly worse, with BiT-small and ConvNeXt-tiny performing better than 96%, the rest of the models' performance is relatively mediocre.

For image recognition with fire, traditional CNNs perform very well, with ResNetV2-50 reaching 98% accuracy and BiT-small even reaching 99%. However, a high true detection rate comes at the cost of a high false alarm rate. Both models have a high false alarm rate, ResNetV2-50 was found to have a false alarm rate of 6.99%. On the contrary, the transformer-based models have a very low false alarm rate, although the accuracy is slightly lower than that of the CNNs models. ConvNeXt-tiny, a member of the traditional CNNs, has a slightly lower accuracy than other CNN models but has a comparable false alarm rate with the transformer-based models.

4.2.3 Detection Latency

Detection latency is a way to describe the time from the onset of the fire until the model raises an alarm and it is measured in frames. Nine datasets are selected to calculate the detection latency, which does not overlap with the data in the test dataset. The result is shown in Table 2.

In Table 2, BiT-small demonstrates the best performance, exhibiting a remarkable ability to recognize small features. However, this also leads to a higher susceptibility to similar features, which is related to BiT-small's higher false alarm rate in Table 1. The performance of ConvNeXt-tiny and DeiT-tiny in the detection latency test is moderate, despite their good inference speed and accuracy. Moreover, the difference in detection latency between them is insignificant. On the other hand, Swin Transformer-tiny is slower in detection latency.

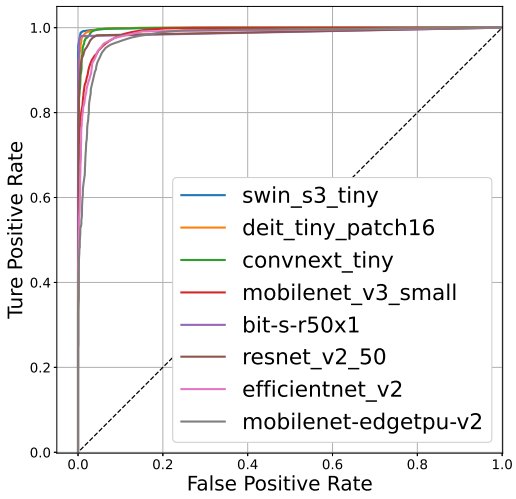

Figure 4: ROC of models. Models are arranged by AUC size from top to bottom.

4.2.4 Implement Latency

Model	Sub-image	Whole Image
Swin Transformer-tiny	0.02227	1.0023
DeiT-tiny	0.02272	1.0223
ConvNeXt-tiny	0.06151	2.7680
MobileNetV3-small	0.00591	0.2660
BiT-small	0.01636	0.7363
ResNetV2-50	**0.00683**	**0.3074**
EfficientNetV2	0.01441	0.6483
Mobilenet-edgetpu-v2	0.00892	0.4013

Table 3: The time it takes for the model to process the image to make a prediction. Measured in seconds.

In addition to the model's ability to detect fires at the earliest stages, we are also interested in image processing speed. Implement latency is used to measure the time that a model needed to process a single image. Since we split our single image into 45 sub-images, we both calculate the time of processing a single image and a sub-image. At the same time, Table 3 indicates the speed of models and we can note that MobileNetV3-small has the shortest implement latency and ResNetV2-50 has the second shortest implement latency. While ConvNeXt-tiny has a smaller false alarm rate compared to Swin-tiny and DeiT-tiny, it is not as efficient in processing images as these transformer-based models.

5 Conclusion

In this paper, we chose 8 pre-trained models that are widely used in image classification to validate the effectiveness and performance of artificial deep neural networks for wildfire detection using regular visible-range cameras. Transfer learning is used to train these models and the models are evaluated based on different indicators. Based on the experiments summarized above, the strengths and limitations of these models are determined. According to the criteria of traditional image classifiers, transformer-based models Swin Transformer and DeiT achieved the highest AUC and accuracy. Traditional CNNs models have slightly lower AUC and accuracy. On the other hand, traditional CNN models except ConvNeXt are more efficient in implementation and superior in detecting tiny smoke features. Swin Transformer and DeiT consume more time to implement relatively, but it is still acceptable in practice because the processing time of an entire image frame size is less than 1 second except for Swin Transformer and DeiT networks. Unfortunately, Swin Transformers and Deit are less sensitive to tiny smoke features. ConvNext-tiny has the lowest false alarm rate. A low false alarm rate is very important for the acceptance of the use of machine learning in wildfire detection problems.

Future research will focus on the use of other types of data, such as thermal and multispectral imagery. Additionally, the development of hybrid models that combine deep learning algorithms with traditional machine learning techniques and domain knowledge can also be a promising direction for improving the accuracy of wildfire detection systems.

Declarations

Ethics approval Not applicable.

Funding This work was supported by National Science Foundation (NSF) under grant 1934915 and the University of Illinois Chicago Discovery Partners Institute Seed Funding Program.

Competing interests The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Availability of data and materials All the testing data are available upon request by contacting the corresponding author and on

HPWREN wildfire dataset[36].

Authors' contributions Zilinag Hong carried out the experiments. Ziliang Hong and Emadeldeen Hamdan wrote the main manuscript text. Zilinag Hong, Yifei Zhao and Tianxiao Ye collected the dataset. Hongyi Pan and Ahmet Enis Cetin supervised the work. All authors reviewed the manuscript.

References

[1] B.U. Toreyin, Y. Dedeoglu, A.E. Cetin, in *2005 13th European signal processing conference* (IEEE, 2005), pp. 1–4

[2] B.U. Toreyin, Y. Dedeoglu, U. Gudukbay, A.E. Cetin, Computer vision based method for real-time fire and flame detection. Pattern recognition letters **27**(1), 49–58 (2006)

[3] F. Yuan, A fast accumulative motion orientation model based on integral image for video smoke detection. Pattern Recognition Letters **29**(7), 925–932 (2008)

[4] B.U. Toreyin, A.E. Cetin, in *2009 IEEE International Conference on Acoustics, Speech and Signal Processing* (IEEE, 2009), pp. 1461–1464

[5] O. Günay, K. Taşdemir, B. Uğur Töreyin, A.E. Çetin, Fire detection in video using lms based active learning. Fire technology **46**, 551–577 (2010)

[6] Y.H. Habiboglu, O. Gunay, A.E. Cetin, in *2011 19th European Signal Processing Conference* (IEEE, 2011), pp. 894–898

[7] Y.H. Habiboglu, O. Gunay, A.E. Cetin, Covariance matrix-based fire and flame detection method in video. Machine Vision and Applications **23**, 1103–1113 (2012)

[8] O. Gunay, B.U. Toreyin, K. Kose, A.E. Cetin, Entropy-functional-based online adaptive decision fusion framework with application to wildfire detection in video. IEEE Transactions on Image Processing **21**(5), 2853–2865 (2012)

[9] A.E. Çetin, K. Dimitropoulos, B. Gouverneur, N. Grammalidis, O. Günay, Y.H. Habiboğlu, B.U. Töreyin, S. Verstockt, Video fire detection–review. Digital Signal Processing **23**(6), 1827–1843 (2013)

[10] O. Günay, A.E. Çetin, in *2015 IEEE International Conference on Image Processing (ICIP)* (IEEE, 2015), pp. 3087–3091

[11] S. Aslan, U. Gudukbay, B.U. Toreyin, A.E. Cetin, in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (IEEE, 2019), pp. 8315–8319

[12] P. Jindal, H. Gupta, N. Pachauri, V. Sharma, O.P. Verma, in *Soft Computing: Theories and Applications: Proceedings of SoCTA 2020, Volume 2* (Springer, 2021), pp. 539–550

[13] A.E. Cetin, B. Merci, O. Gunay, B.U. Toreyin, S. Verstockt, *Methods and techniques for fire detection: signal, image and video processing perspectives* (Academic Press, 2016)

[14] H. Pan, D. Badawi, X. Zhang, A.E. Cetin, Additive neural network for forest fire detection. Signal, Image and Video Processing **14**, 675–682 (2020)

[15] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2016), pp. 779–788

[16] Q.x. Zhang, G.h. Lin, Y.m. Zhang, G. Xu, J.j. Wang, Wildland forest fire smoke detection based on faster r-cnn using synthetic smoke images. Procedia engineering **211**, 441–446 (2018)

[17] X. Wu, X. Lu, H. Leung, in *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)* (IEEE, 2017), pp. 1954–1959

[18] H. Pan, D. Badawi, A.E. Cetin, Computationally efficient wildfire detection method using a deep convolutional network pruned via fourier analysis. Sensors **20**(10), 2891

(2020)

[19] K. He, X. Zhang, S. Ren, J. Sun, in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14* (Springer, 2016), pp. 630–645

[20] G. Hinton, O. Vinyals, J. Dean, Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015)

[21] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, H. Jegou, in *International conference on machine learning* (PMLR, 2021), pp. 10,347–10,357

[22] M. Tan, Q. Le, in *International conference on machine learning* (PMLR, 2021), pp. 10,096–10,106

[23] A. Kolesnikov, L. Beyer, X. Zhai, J. Puigcerver, J. Yung, S. Gelly, N. Houlsby, in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16* (Springer, 2020), pp. 491–507

[24] A. Howard, M. Sandler, G. Chu, L.C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, et al., in *Proceedings of the IEEE/CVF international conference on computer vision* (2019), pp. 1314–1324

[25] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, B. Guo, in *Proceedings of the IEEE/CVF international conference on computer vision* (2021), pp. 10,012–10,022

[26] Z. Liu, H. Mao, C.Y. Wu, C. Feichtenhofer, T. Darrell, S. Xie, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 11,976–11,986

[27] K. He, X. Zhang, S. Ren, J. Sun, in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2016), pp. 770–778

[28] A.G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam, Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861 (2017)

[29] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L.C. Chen, in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2018), pp. 4510–4520

[30] M. Tan, Q. Le, in *International conference on machine learning* (PMLR, 2019), pp. 6105–6114

[31] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need. Advances in neural information processing systems **30** (2017)

[32] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al., An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)

[33] N. Grammalidis, K. Dimitropoulos, E. Cetin, Firesense database of videos for flame and smoke detection. IEEE Trans Circuits Syst Video Technol **25**, 339–351 (2017)

[34] J. Deng, W. Dong, R. Socher, L.J. Li, K. Li, L. Fei-Fei, in *2009 IEEE conference on computer vision and pattern recognition* (Ieee, 2009), pp. 248–255

[35] K. Woods, K.W. Bowyer, Generating roc curves for artificial neural networks. IEEE Transactions on medical imaging **16**(3), 329–337 (1997)

[36] A. University of California San Diego, California. The high performance wireless research and education network. (2019). Accessed December 25, 2022