FireNet-Tiny: Very-Low Parameter Count High Performance Fire Detection Model

Olalekan Joshua Oyebanji, Stefy Oliver, Chukwuka Ekezie Ogonna, Asra Aslam, and Mohammad Samar Ansari

Abstract In daily life, fire threats result in significant costs on the ecological, social, and economic levels. It is essential to outfit the assets with fire prevention systems due to the sharp rise in the frequency of fire mishaps. To prevent such mishaps, several studies have been conducted to develop optimal and potent fire detection models. While the earliest methods were thermal/chemical in nature, image processing was later applied for identification of fire. More recent methods have taken advantage of the significant advancements in deep learning models for computer vision. However, in order to maintain a suitable inference time (leading towards real-time detection) and parameter count, the majority of deep learning models have to make trade-offs between their detection speed and detection performance (accuracy/recall/precision). The very lightweight convolution neural network we offer in this paper is specifically designed for the fire detection use case. The proposed model can be embedded in real-time fire monitoring equipment and could also prove potentially useful for future fire monitoring methods such as unmanned aerial vehicles (drones). By further diminishing the trainable parameter count of the model, the fire detection results obtained using the proposed FireNet-Tiny significantly outperform the prior low parameter count models. When tested against FireNet dataset, FireNet-Tiny, which only comprises 261,922 parameters, was shown to have an overall accuracy of 95.75%. In comparison, FireNet-*v2* provided 94.95% accuracy with 318,460 parameters.

O. J. Oyebanji · S. Oliver · C. E. Ogonna · M. S. Ansari (✉)
University of Chester, Chester, UK
e-mail: m.ansari@chester.ac.uk

O. J. Oyebanji
e-mail: 2220823@chester.ac.uk

S. Oliver
e-mail: 2227367@chester.ac.uk

C. E. Ogonna
e-mail: 2218114@chester.ac.uk

A. Aslam
Faculty of Medicine and Health, University of Leeds, Leeds, UK
e-mail: a.aslam2@leeds.ac.uk

S. J. Nanda et al. (eds.), *Data Science and Applications*, Lecture Notes in Networks and Systems 821, https://doi.org/10.1007/978-981-99-7814-4_40

Keywords Fire Detection · Deep Learning · Lightweight Models · FireNet

1 Introduction

One of the many risks that endangers human life and destroys property is fire, which is why there are fire hazards. More than ever, early fire detection is crucial to avoiding potentially fatal fire threats. Traditional thermal, photometric, and chemical fire detectors are examples of methods that are currently used to raise the alarm and stop fire accidents. The main disadvantage of these techniques is that they need a significant volume of smoke or fire to detect and activate. Additionally, it has a high probability of false triggering and cannot be used for outside detection in places like forests, streets, playgrounds, industries, etc. Such elaborate fire detectors are being replaced with hand-designed detectors that leverage the advancements in digital imaging and deep learning-assisted visual perception technology along with smart video processing approaches. Initially, colour and shape characteristics of smoke were used to extract hand-designed features for distinguishing smoke and fire in picture and video categorization. Due to technological progress, researchers have introduced image processing, computer vision, and artificial intelligence (AI) methods that address challenges associated with previous methodologies. The performance of fire detectors using these cutting-edge technology is more resilient, prompt/quick, and reliable. Furthermore, because the accumulation of smoke is not a necessary condition for such a detector, such detectors can also operate in open-air settings exhibiting fewer false triggers and quicker inference times.

With the widespread presence of closed-circuit television (CCTV), or some other form of camera, in almost every location, including streets, highways, businesses, shopping centres, parks, the infrastructure is extensively interconnected. This networking allows for comprehensive visualization of objects and activities through cameras and CCTVs, which is further complemented by monitoring using the Internet of Things (IoT). As a result of this integration, visual-based fire detectors have facilitated the convergence of cutting-edge vision-based fire detection technology with surveillance systems [1]. This has the benefit of the elimination of the hardware/materials (and therefore, cost) used in thermal/chemical fire detectors by simply employing a software construct (the AI model) in real-time image/video-based systems for fire detection. Modern fire detectors frequently use deep learning (DL) models, which do away with the requirement for human feature extraction and enable automatic feature extraction directly from images or frames extracted from videos in real-time [2]. Furthermore, studies have demonstrated that deep learning-based fire detection systems utilizing visual data exhibit higher accuracy, reduced false alarm rates, enhanced robustness, and increased reliability.

Originality and Contribution: In this paper, we highlight an enhanced version of FireNet[3] and FireNet-*v2* [4], with substantially reduced trainable parameter count and reduced false triggering (improved recall/sensitivity) as compared to the predecessors. The proposed lightweight model's uniqueness comes from the fact that it was created specifically with fire detection in mind. We then go on to demonstrate that the suggested model is still able to outperform its predecessors despite having a substantially smaller number of parameters.

A layout of the paper's contents is provided below. Section-II contains a brief but pertinent discussion about some earlier work on manually created features and AI methods for fire detection. Section-III contains details of the proposed DL model's architecture and the rationale for adopting FireNet-Tiny over FireNet and FireNet-*v2*. Section-IV offers a discussion/analysis of the results and also includes a comparison with other published works. The last section, Section-V, contains some concluding remarks.

2 Related Works

In the pursuit of mitigating fire hazards, significant research has been dedicated to developing robust and efficient fire detection solutions. Contemporary fire detection systems commonly rely on ion-based, infrared, or optical sensors that require proximity to the source or location to trigger activation. However, these systems may not be suitable for certain environments such as markets, schools, and open areas. As a preferred substitute for these sensors, vision-based sensors provide several advantages over traditional sensors, including lower costs, quicker response times, and a wider area of surveillance coverage [5], to name just a few. They do, however, have some serious drawbacks, such as a reliance on scene complexity, varying lighting, and varying image quality. In the initial stages, researchers primarily concentrated on the motion and colour components of flame detection. They devised custom algorithms for fire detection [6–8], which were engineered manually. These detection methods are computationally lightweight and can be easily deployed on resource-limited embedded devices like the NVIDIA Jetson Nano and the Raspberry Pi. They exhibit satisfactory performance on the evaluation parameter of frame rates. However, a drawback lies in their dependence on manual (human involved) extraction of usable features from fire-containing videos/images. Consequently, manual feature engineering becomes time-consuming and often yields unsatisfactory outcomes, particularly when dealing with datasets containing numerous images.

Considering this situation, manually crafted vision-based fire detection systems are giving way to DL-based approaches. These techniques have increasingly substituted less accurate counterparts (i.e. ones with higher false triggering rates). They demonstrate superior performance across various parameters. Deep learning approaches bring the advantage of autonomously extracting significant features from presented data. This leads to a more efficient process that relies less on human intervention. These approaches significantly advance the state-of-the-art in procedures like object recognition and classification [2, 9]. Numerous deep learning algorithms have been proposed for fire detection, as documented in technical literature [9–15]. However, these research models prove unsuitable for real-world fire detection applications where cost-effective, resource-constrained hardware is typically the only available option.

Recently, there has been a substantial amount of academic work focused on creating lightweight deep learning models intended for use on edge devices with limited resources [16]. Notable instances of such compact deep learning models are evident in the realms of automated identification for various medical conditions [17], forgery identification in images [18], and a multitude of other real-life applications [16]. Some pertinent models for fire detection in varied situations include, but are not limited to, [19–26]. While most of these works are quite novel and effective, the results in these published works are presented on several different datasets. Later on in this paper, for the comparison of the performance of the proposed model, we shall only be considering works which have presented their test results on the FireNet dataset, which has been utilized in this work.

3 Proposed Model

The enormous success of the application of the Convolutional Neural Networks (CNN) to different computer vision tasks prompted researchers to apply CNNs for the task of fire detection from images and videos. CNNs are typically comprised of three types of layers. The convolutional layer, which applies convolutions to the input data, is a crucial part of a CNN. In order to incorporate nonlinearities into the network, the convolutional layer is often followed by nonlinear activation functions like rectified linear unit (ReLU). The network can better describe complex interactions between input and output because to these activation functions. Pooling layers, like max pooling or average pooling, are also used by CNNs to minimize the spatial dimensions of the data, improving the network's computing efficiency and decreasing its sensitivity to minute spatial fluctuations. By combining layers, one can capture the most crucial details while excluding some of the less critical ones. Typically, a CNN's final component is made up of one or more *fully connected* layers with optional dropouts in between (referred to as *Dense* layers in Keras implementations). Overall, CNNs are quite effective for a variety of computer vision tasks because of their hierarchical nature, which enables them to automatically acquire increasingly complicated and abstract properties from raw pixel input.

As discussed in the previous section, we have observed several variations of CNNs being utilized to enhance fire detection accuracy and reduce false alerts (incorrect predictions), driven by advancements in fog/edge computing set-ups and the potential of intricate feature identification [10, 12–15, 27–29]. However, the high-parameter-count structure of such CNN-based approaches makes them challenging to harness on affordable computing hardware. To address this, we introduce FireNet-Tiny, a lightweight (low-parameter-count) neural network. Despite having significantly fewer parameters than its predecessors (FireNet [3] and FireNet-*v2* [4]), FireNet-Tiny sustains its fire identification performance.

3.1 Motivation for FireNet-Tiny

The choice of a model has always been crucial for various applications. A quick and precise model is essential in these situations, because even small delays can cause large losses in terms of people and money. In the beginning, FireNet was designed to be superior to other (then) available CNN models. Its shallowness compared to earlier DL-based fire identification approaches, which are generally massive convolutional neural networks capable of detecting fires in real-time at a speed of at least 24 frames per second (or maybe greater), was its main advantage. It worked effectively in real-time (continuous) fire detection applications and was also suitable for embedded and mobile ones. The model runs on resource-constrained, reasonably priced single-board computing platforms (e.g. Raspberry Pi) with a frame rate of more than 24 frames per second. FireNet-*v2* was proposed as a further improvement over FireNet with better performance metrics and lower parameter count.

FireNet-Tiny is driven by the goal of reducing its model size in two crucial aspects: (i) minimizing the necessary computations and (ii) lowering the count of trainable parameters. This optimization results in improved accuracy compared to its predecessors. This paper will provide a comprehensive explanation, detailing how FireNet-Tiny surpasses the performance of both the original FireNet and the enhanced FireNet-*v2*, all while maintaining a decreased parameter count. Specifically, the advancements we have made over the previous versions are outlined as follows:

- While the number of convolutional layers in both FireNet and FireNet-*v2* were three, the proposed FireNet-Tiny employs an additional convolutional layer (total of 4 layers).
- FireNet-Tiny employs only 3 fully connected (Dense) layers, as opposed to 4 in the previous iterations. The most significant benefit is the very-low-parameter count (261,922) network.
- ‘ReLU’ is used universally as the activation function of choice across the entire network, except the last Dense layer for which a Softmax activation is required due to the desired output type being of 2 classes.

3.2 FireNet-Tiny Architecture

In FireNet-Tiny, we employed the initial (input) convolutional layer accepting image sizes of 64 $\times$64 $\times$3. The sequentially ordered arrangement of intermediate layers includes convolution, average pooling, and dropout layers. These layers employ the rectified linear unit (ReLU) activation function. Each of the four convolutional layers is paired with average pooling. We’ve employed 16 filters in the initial layer, 32 filters in the second, 64 filters in the third, and 128 filters in the fourth. Furthermore, we’ve applied varying dropout rates (illustrated in Fig. 1) to all four Conv layers. The

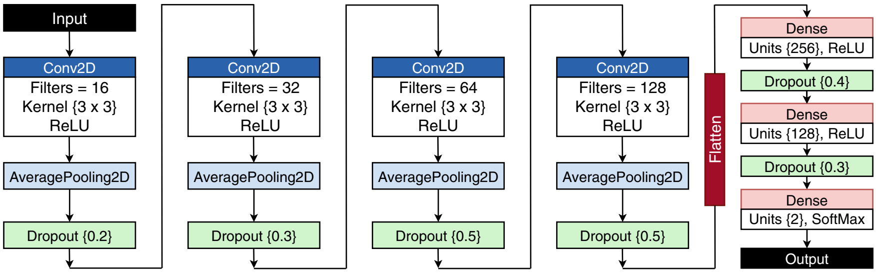

Fig. 1 Layer-wise structure of the designed FireNet-Tiny

kernel size remains fixed at 3×3 for each Conv layer. Subsequent to the Conv layers, a Flatten layer is followed by two Dense layers, containing 256 and 128 neurons, respectively, all utilizing the 'ReLU' activation function. The prediction layer, a fully connected layer (Dense layer) utilizing a 'Softmax' output, comprises dual neurons and provides the outputs of 'non-fire' and 'fire' detections. At any point, only one of these signals will be predominantly active. The detailed layer-wise architecture is shown in Fig. 1 which is made up of 11 layers (dropout not counted). Overall, there are only a miniscule 261,922 trainable parameters.

4 Training Results

Next, we provide details on the data used, the tests run, and the outcomes.

4.1 The Data

In the past decade, DL models designed for computer vision tasks have showcased exceptional performance across numerous practical scenarios. However, the performance is greatly influenced by the both the quality and quantity of data available for the model training. There are not enough high-quality datasets available for DL models for fire detection. The dataset created for FireNet [3], which includes 46 movies of fire scenarios and 16 non-fire movies (i.e. where fire was not present),[1] was used to train and test the FireNet-Tiny model This dataset is available as an open-source resource. Overall, 1124 fire images and 1301 non-fire images were used in this work. Despite the data samples in the FireNet dataset being limited in count, they contain a very diverse range of realistic images, making them more challenging for the model to learn from. This assertion will be confirmed in a subsequent part of this

[1] https://github.com/arpit-jadon/FireNet-LightWeight-Network-for-Fire-Detection.

manuscript, where it is demonstrated that for the unseen samples downloaded from the Internet, the accuracy provided by the FireNet-Tiny model is not compromised.

4.2 Training Details

The training and test/prediction stages are illustrated in Fig. 2. The several hyper-parameters and transfer functions we used for FireNet-Tiny are listed in Table 1. Due to the model's design, which yields outputs in the form of 'one-hot' encoding involving two outputs, with only one being dominant for each test image (fire or non-fire), the final layer was implemented with the Softmax activation function. This choice was made over the Sigmoid activation consciously.[2] The utilization of Softmax activation compels the model to diminish the computed probabilities of non-desired classes (and vice versa, i.e. to enhance the estimated probability of desired class). This occurs because Softmax guarantees that the total of designated probabilities for both output classes sums to unity. As a result, there is a more distinct classification of fire or non-fire scenarios utilizing this method. If a Sigmoid activation function had been utilized, there would have been just one output, displaying the likelihood that the given image included fire or not. With a batch size equal to 32, we used the optimizer Root Mean Square Propagation (RMSprop) for the training of the model over 100 epochs.

The hardware utilized for the training of FireNet-Tiny was quite unexceptional with an Intel® Core™ i5-8265U CPU@1.80GHz with 8 GB of RAM running Windows 10 Pro. Keras 2.10.0 and Python 3.10 installed in a 64-Bit Anaconda Navigator environment were used for model training and testing.

4.3 Performance Evaluation

The FireNet-Tiny model was first trained and then evaluated on the FireNet dataset. The predictions are divided into Correct Detection/True Positive (TP), Type-II Error/False Negative (FN), Type-I Error/False Positive (FP), and Correct Rejection/True Negative (TN), which are also illustrated in the form of a confusion matrix in Fig. 3 from where it can be confirmed that FireNet-Tiny does well on the dataset, with the proportions of both the Type-I and Type-II errors being low. Table 2 presents a comparison of the proposed model with its predecessors, from where it can be ascertained that FireNet-Tiny indeed outperforms the others on accuracy, recall, and F1-Score metrics, while having a lesser parameter count. Figure 4 shows several examples of the photos that FireNet-Tiny correctly classified. Similar to that, Fig. 5 shows two examples of photos that FireNet-Tiny was unable to classify correctly.

[2] Opting for Softmax over Sigmoid does not result in any degradation in performance since these two functions essentially serve the same purpose in binary classification.

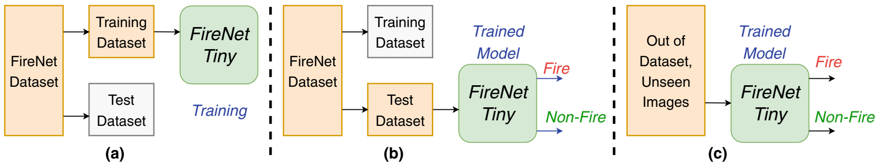

Fig. 2 **a** Model fitting and **b** evaluation stages of FireNet-Tiny on the FireNet dataset; **c** testing the FireNet-Tiny on unseen images from the Internet

Table 1 Characteristics of the FireNet-Tiny DL model

Parameter	Value
(a) Parameter counts	
Parameter count	261,922
Batch size	32
Epoch	100
Validation split	0.1
Loss	Sparse categorical
	Crossentropy
Optimizer	RMSprop
Layer	Activation
(b) Activation functions	
1st layer (Conv)	ReLU
2nd layer (Conv)	ReLU
3rd layer (Conv)	ReLU
4th layer (Conv)	ReLU
Dense layer(s)	ReLU
Output layer	Softmax

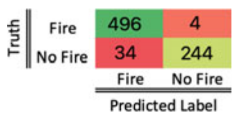

Fig. 3 Error matrix for the FireNet-Tiny DL model on the FireNet data

Table 2 Performance comparison of FireNet-Tiny with FireNet-*v2* and FireNet

Models Metrics	FireNet-Tiny (%)	FireNet v2 (%)	FireNet (%)
Accuracy	**95.75**	94.95	93.91
Recall	**96.62**	93.25	94
Precision	97.11	**99.28**	97
F1-Score	**96.8**	96.17	95
No of parameters	**261,922**	318,460	646,818

The best entries are emphasized in bold

Fig. 4 Samples of images which were correctly identified as images with or without fire present

Fig. 5 Examples of incorrectly classified images using FireNet-Tiny on the test subset of the FireNet dataset

4.3.1 Comparison with Existing Works

Subsequently, we offer a juxtaposition between the proposed FireNet-Tiny and currently established state-of-the-art fire identification systems with high performance. As stated below, the most significant feature of the proposed model is the substantially shallow (low number of layers) architecture and, as a result, the meagre parameter count (only 261,922). At this juncture, it is necessary to admit that technologies for improved performance fire detection exist and are easily available in the literature. The deployment of such corresponding technical papers into actual commercial use is, however, constrained by their heavyweight architectures (large parameter counts, high inference times). The findings of comparisons of the proposed models with other lightweight models are therefore considered here.

This section encompasses the outcomes of quantitative assessments, drawing comparisons between the existing methodologies and the present study. To the best of our understanding, Table 3 contrasts the accuracy and parameter counts among several recent publications related to fire detection that employ the same dataset [1, 30–32]. These test results are based on the inferences provided by the trained model over the *Test* subset from the FireNet dataset using 778 images. Additionally, the performance assessment of FireNet-Tiny is portrayed in Fig. 6, alongside a comparison to

Table 3 Prediction accuracy comparison of FireNet-Tiny with several existing models trained on the same data as used in this work

Model Reference	Prediction Accuracy (%)	Parameter #
This Work	95.7	261922
[4]	94.9	318460
[3]	93.9	646818
[1]	96.6	≈ 50 Million
[30]	96.3	956226
[31]	93.6	23482

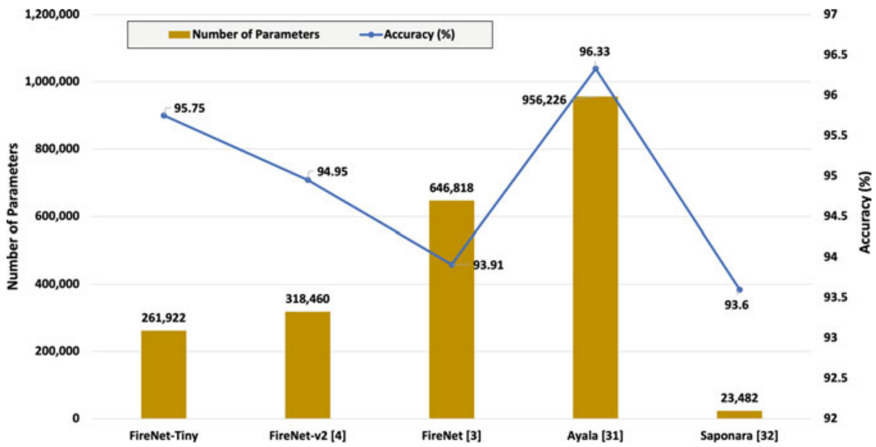

Fig. 6 Illustration of prediction accuracy and parameter counts of FireNet-Tiny with existing similar models trained over the same data

existing counterparts with reported test outcomes on the FireNet dataset. The data presented in Fig. 6 clearly indicates that FireNet-Tiny achieves a superior level of accuracy while maintaining a much lower parameter count. It needs to be mentioned that for [1] the number of parameters are not provided in the published work, and the parameter count provided in Table 3 is an estimated value considering that the fire detection model in [1] uses YOLOv2.

4.3.2 Performance on Unseen Images

Lastly, to ascertain the classification performance of the proposed model over unseen (new) images, 200 random images (100 with fire, 100 without fire) were downloaded from the Web.[3] The *trained* FireNet-Tiny model was used to make inferences on these 200 images, and the model resulted in an accuracy of 84.5% with an F1-Score of

[3] https://github.com/Asra-Aslam/UnseenNet.

Fig. 7 Some examples of 'NoFire' images collected from the Internet which are quite similar to 'fire' images—these affected the overall classification accuracy of the FireNet-Tiny model

87%. The most significant observation was that the recall was 100%—which goes on to show that the trained FireNet-Tiny model did not miss any 'fire' situation. Further, exploration about the lower-than-expected accuracy over the 200 Internet images led to the discovery that several images in the 'NoFire' class were quite similar in appearance to the 'fire' images (some such examples are shown in Fig. 7). The model classified these as 'fire' although these are 'NoFire' images.

5 Conclusion

In this paper, a lightweight deep learning model with an exceptionally low parameter count was introduced for image classification into fire and non-fire categories. The newly proposed FireNet-Tiny is an enhanced iteration of FireNet [3] and FireNet-*v2*, and features notably fewer trainable parameters than its precursors. The proposed model was evaluated on the FireNet dataset, and the estimated inference accuracy of FireNet-Tiny surpassed that of currently available models with low parameter counts. As an example, FireNet-Tiny yielded an accuracy of 95.75% with only 261,922 parameters in comparison with FireNet's 96.53% accuracy (646,818 parameters) [3] and FireNet-*v2*'s 94.95% accuracy (318,460 parameters) [4].

References

1. Sergio Saponara, Abdussalam Elhanashi, Alessio Gagliardi, Real-time video fire/smoke detection based on cnn in antifire surveillance systems. J. Real-Time Image Proc. **18**(3), 889–900 (2021)
2. S.S.A. Zaidi, M.S. Ansari, A. Aslam, N. Kanwal, M. Asghar, B. Lee, A survey of modern deep learning based object detection models. Digit. Signal Process 103514 (2022)
3. A. Jadon, M. Omama, A. Varshney, M.S. Ansari, R. Sharma, Firenet: a specialized lightweight fire and smoke detection model for real-time iot applications. *arXiv preprint* arXiv:1905.11922 (2019)
4. A. Shees, M.S. Ansari, A. Varshney, M.N. Asghar, N. Kanwal, Firenet-v2: Improved lightweight fire detection model for real-time iot applications. Procedia Comput. Sci. **218**, 2233–2242 (2023)

5. K. Dimitropoulos, P. Barmpoutis, N. Grammalidis, Spatio-temporal flame modeling and dynamic texture analysis for automatic video-based fire detection. IEEE Trans. Circ. Syst. Video Technol. **25**, 339–351 (2015)
6. T.-H. Chen, P.-H. Wu, Y.-C. Chiou, An early fire-detection method based on image processing. Int. Conf. Image Proc. **2004**(3), 1707–1710 (2004)
7. T. Çelik, H. Özkaramanlı, H. Demirel, Fire and smoke detection without sensors: image processing based approach. 1794–1798 (2007)
8. A. Rafiee, R. Dianat, M. Jamshidi, R. Tavakoli, S. Abbaspour, Fire and smoke detection using wavelet analysis and disorder characteristics, in *2011 3rd International Conference on Computer Research and Development*, vol. 3. (IEEE, 2011), pp. 262–265
9. A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks. Adv. Neural Inf. Processing Syst. **25** (2012)
10. Q. Zhang, J. Xu, L. Xu, H. Guo, Deep convolutional neural networks for forest fire detection (2016)
11. J. Sharma, O.-C. Granmo, M. Goodwin, J. T. Fidje, Deep convolutional neural networks for fire detection in images, in *in International Conference on Engineering Applications of Neural Networks*, pp. 183–193 (2017)
12. K. Muhammad, J. Ahmad, S.W. Baik, Early fire detection using convolutional neural networks during surveillance for effective disaster management. Neurocomputing, **288**, 30–42 (2018)
13. K. Muhammad, J. Ahmad, Z. Lv, P. Bellavista, P. Yang, S.W. Baik, Efficient deep cnn-based fire detection and localization in video surveillance applications. IEEE Trans. Syst. Man Cybern. Syst. **49**(7), 1419–1434 (2018)
14. K. Muhammad, J. Ahmad, I. Mehmood, S. Rho, S.W. Baik, Convolutional neural networks based fire detection in surveillance videos. IEEE Access **6**, 18174–18183 (2018)
15. K. Muhammad, S. Khan, M. Elhoseny, S.H. Ahmed, S.W. Baik, Efficient fire detection for uncertain surveillance environment. IEEE Trans. Ind. Inf. (2019)
16. C.-H. Wang, K.-Y. Huang, Y. Yao, J.-C. Chen, H.-H. Shuai, W.-H. Cheng. Lightweight deep learning: an overview. IEEE Consum. Electron. Mag. (2022)
17. Shamik Tiwari, Anurag Jain, A lightweight capsule network architecture for detection of covid-19 from lung ct scans. Int. J. Imaging Syst. Technol. **32**(2), 419–434 (2022)
18. M.N. Abbas, M.S. Ansari, M.N. Asghar, N. Kanwal, T. O'Neill, B. Lee, Lightweight deep learning model for detection of copy-move image forgery with post-processed attacks, in *2021 IEEE 19th World Symposium on Applied Machine Intelligence and Informatics (SAMI)* (IEEE, 2021), pp. 000125–000130
19. Y. Xing, L. Zhong, X. Zhong, An encoder-decoder network based fcn architecture for semantic segmentation. Wirel. Commun. Mob. Comput. (2020)
20. J. Zhang, H. Zhu, P. Wang, X. Ling, Att squeeze u-net: a lightweight network for forest fire detection and recognition. IEEE Access (2021)
21. M.A. Akhloufi, R.B. Tokime, H. Elassady, Wildland fires detection and segmentation using deep learning. pattern recognition and tracking. xxix. Int. Soc. Opt. Photonics Proc. SPIE 2018, 10649, 106490B (2018)
22. V.S. Bochkov, L.Y. Kataeva, Wuunet: advanced fully convolutional neural network for multi-class fire segmentation. Symmetry (2021)
23. Xu. Renjie, Haifeng Lin, Lu. Kangjie, Lin Cao, Yunfei Liu, A forest fire detection system based on ensemble learning. Forests **12**(2), 217 (2021)
24. G. Jocher, A. Chaurasia, A. Stoken, J. Borovec, NanoCode012, Y. Kwon, T. Xie, J. Fang, Imyhxy, K. Michael, A.V. Lorna, D. Montes, J. Nadar, Laughing, tkianai, yxNONG, P. Skalski, Z. Wang, A. Hogan, C. Fati, L. Mammana, AlexWang1900, D. Patel, D. Yiwei, F. You, J. Hajek, L. Diaconu, M.T. Minh. Ultralytics/yolov5: v6.1 - TensorRT, TensorFlow Edge TPU and OpenVINO Export and Inference, Feb (2022)
25. M. Tan, R. Pang, Q.V. Le, Efficientdet: Scalable and efficient object detection, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2020), pp. 10781–10790

26. M. Tan, Q. Le, Efficientnet: rethinking model scaling for convolutional neural networks 6105–6114, (2019)
27. K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition. *arXiv preprint* arXiv:1409.1556 (2014)
28. K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778 (2016)
29. Pasquale Foggia, Alessia Saggese, Mario Vento, Real-time fire detection for video-surveillance applications using a combination of experts based on color, shape, and motion. IEEE Trans. Circ. Syst. Video Technol. **25**(9), 1545–1556 (2015)
30. A. Ayala, E. Lima, B. Fernandes, B.L.D. Bezerra, F. Cruz, Lightweight and efficient octave convolutional neural network for fire recognition. 1–6 (2019)
31. S. Saponara, A. Elhanashi, A. Gagliardi, Exploiting r-cnn for video smoke/fire sensing in antifire surveillance indoor and outdoor systems for smart cities, in *2020 IEEE International Conference on Smart Computing (SMARTCOMP)* (IEEE, 2020), pp. 392–397
32. A.H. Altowaijri, M.S. Alfaifi, T.A. Alshawi, S.A. Alshebeil, A privacy-preserving iot-based fire detector. IEEE Access 99