The new dataset that contains the forest images with and without fire is used to evaluate the proposed lightweight CNN model. The model proposed is created by testing different hyperparameter settings with the consideration of the model's lightness. The best performing CNN model proposed has convolution with three filters in the first and 32 filters for the rest of the layers. To ensure that the model presented is not overfitting, 10-fold cross validation is performed. In the preliminary tests (a single pass with mixed train-test split), the model reaches to an accuracy of 99.12%. For single pass and 10-fold cross validation processes, stochastic gradient descent is used as an optimizer. RELU is used as an activation function. Each layer contains a dropout at the rate of 0.25. The model is trained for 100 epochs with a learning rate of 0.01 using cross-entropy loss, and early stopping with patience of 10 epochs. In other words, if the model no longer improves for 10 consecutive epochs, the training process is terminated and latest version of the model is saved.

The representation used for the captured images is critical for the whole system as it directly affects the complexity of the model. Complexity, in turn, specifies the computational and storage resources required in smart sensors. Although the input sizes can be reduced, the detection accuracy must be at acceptable levels. In the proposed framework, the images are resized to 64 $\times$ 64 $\times$ 3 to reduce the computational cost. This reduction in the input size does not significantly affect the learning process, since the proposed framework mainly focuses on early detection. It is sufficient to use the input for classification as fire or non-fire. Other features needed for localization of the fire or for the detection of other factors such as the direction of the fire, are not necessary for early detection. While the input size is reduced as explained, the model proposed in this study comes with a detection accuracy of 98.28%. There are similar studies in the literature that use well-known models, such as References [21, 36, 43] (GoogLeNet architecture) with fire detection accuracies of around 92%, 94.39%, and 99%, respectively. Compared to these approaches, our proposed model works almost as accurately as the best performing model, but requires reduced amounts resources in terms of computation and storage.

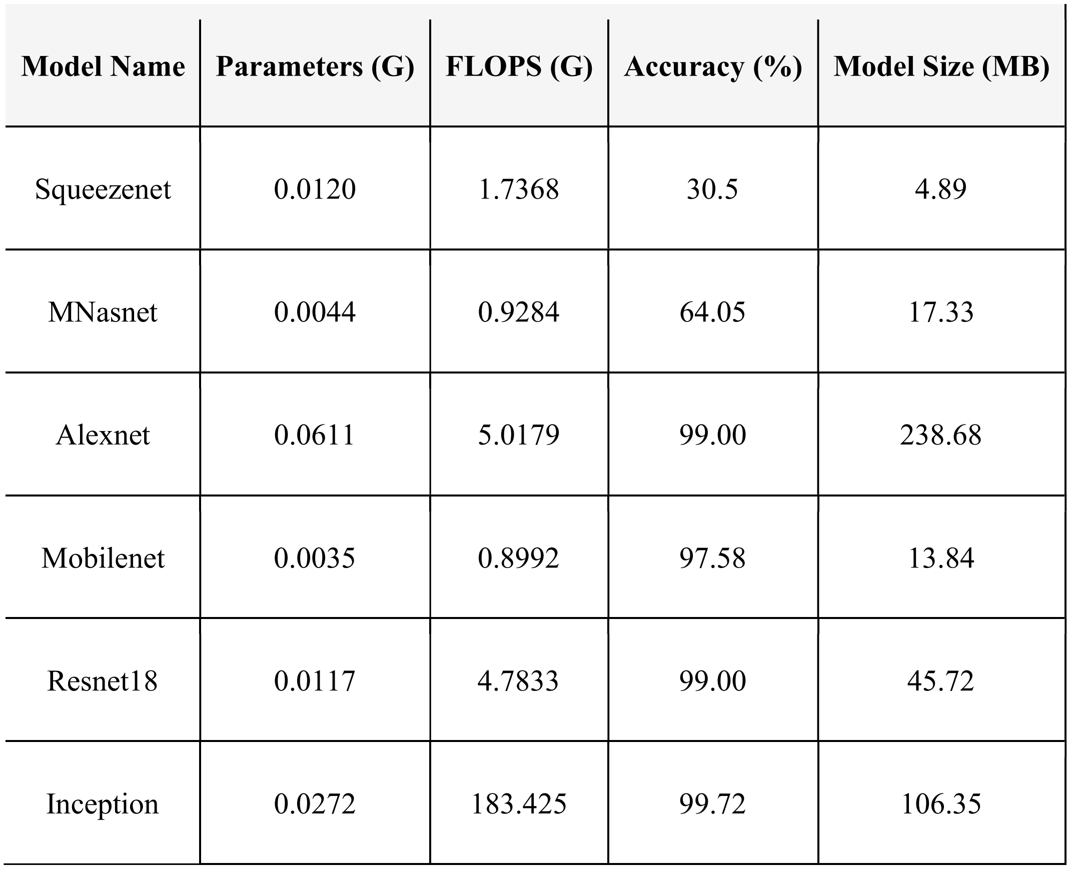
When the characteristics of the model we propose are compared with the characteristics of the models used in existing machine learning-based fire detection systems in more detail, we see that only a few of the studies such as Reference [20] focus on forest fire detection.

Studies such as References [21, 36, 43] mainly consider fire detection in various environments. Similarly, while Reference [20] uses SVM and KNN as machine learning algorithms, References [21, 36, 43] use CNN-based approaches. In Reference [21], RNNs are employed in addition to the CNN-based architectures. All of these studies mainly focus on detection accuracy with well-known architectures such as GoogLeNet. They do not consider the QoS or energy efficiency related issues of the underlying systems. Instead, in our proposed system, by using a hierarchical approach, we can improve the underlying infrastructure in terms of QoS and energy efficiency. The size of the employed machine learning architecture can be quite critical to make sure that we can use the presented architecture in various types of multimedia-enabled smart devices. The model size of the deep learning architecture used is provided only in Reference [43] as 238 MB. The CNN architecture proposed in this study comes with a model size of 1.4 MB. Therefore the improvement in terms of edge computability is quite significant.

To show that the proposed model is lightweight, it is compared to well-known CNN architectures in terms of **float-point operations (FLOP)** [6]. While the proposed model have less than 0.55 GFLOPs, the majority of well-known models have more than 1 GFLOPs in terms of computational requirements. For example, RESNET-50 has about 4 GFLOPs and VGG-13, 16, and 19 have more than 10 GFLOPs [6]. To further emphasize the efficacy of the proposed model, we have also tested the well known light weight models such as Sufflenet [40], Squeezenet [24], MNasnet [58], Alexnet [34], Mobilenet [53], Resnet [22], and Inception [57] using our forest image dataset.
Table 3 shows the number of parameters, FLOPS, model size, and accuracy for each model comparatively with the model proposed in this study.

Table 3. Comparison of the Proposed Model with other Light Weight Approaches

Model Name	Parameters (G)	FLOPS (G)	Accuracy (%)	Model Size (MB)
Proposed	**0.0040**	**0.5362**	**98.28**	**1.40**
Sufflenet	0.0230	0.4516	98.81	9.02



Model Name	Parameters (G)	FLOPS (G)	Accuracy (%)	Model Size (MB)
Squeezenet	0.0120	1.7368	30.5	4.89
MNasnet	0.0044	0.9284	64.05	17.33
Alexnet	0.0611	5.0179	99.00	238.68
Mobilenet	0.0035	0.8992	97.58	13.84
Resnet18	0.0117	4.7833	99.00	45.72
Inception	0.0272	183.425	99.72	106.35

When the results presented in Table 3 are considered in terms of the numbers of parameters, the MNasnet, and Mobilenet models are comparable to the proposed model. However, the proposed model is superior to these models in terms of accuracy, since Mnasnet and Mobilenet have accuracies of 64.05% and 97.58%, respectively, as opposed to 98.28% accuracy of the proposed model. In terms of the FLOPS, the only model comparable to our proposed model is the Sufflenet. Although the accuracy of the Sufflenet is slightly better than our proposed model, in terms of the model size, while our proposed model is around 1.4 MB, the the Sufflenet model is around 9 MB. When the model size is considered, our proposed model is significantly more lightweight than the rest of the models. In terms of size, the closest model to the one we propose is Squeezenet. However, its size is still more than three times the size of the proposed model (4.89 MB), and its accuracy is significantly low (30.5%).

The size of the model and the required computational resources are critical for the proposed framework as these are the main factors affecting the ability to perform early detection at smart end nodes. The use of deep learning architectures with various types of boards for edge computing is also becoming very popular [7, 23, 52]. However, one of the most significant limitations in using various boards to run lightweight deep learning algorithms is the availability of resources, especially in terms of memory requirements. Therefore, these devices are referred to as “Memory-Constrained Edge Devices” [7]. For example, STM32F2 series comes with up to 120 MHz CPU speed, 1 MB of Flash, and up to 128 kB of SRAM [29], whereas Arduino MEGA 2560 supports clock speeds up to 16 MHz, with 256 KB of flash program memory and 8 KB of SRAM [26]. We, therefore, think that all of the metrics presented for the model evaluation play crucial roles. In terms of FLOPs, the proposed approach is at least 1.6 times better than all other approaches except Sufllenet, which performs similar to ours. The proposed approach is superior to all other models in terms of model size, which can be a significantly limiting factor depending on the type of end devices being used. Our proposed architecture is more than three times smaller than the next smallest architecture (Squeezenet) and more than six times smaller than Sufflenet. Finally, in terms of accuracy, the best models are Inception, Resnet18, and Alexnet with accuracies of 99.72%, 99.00%, and 99.00%, respectively, as opposed to the accuracy of the proposed model, which is 98.28%. Although the accuracy of the proposed model is quite close to the highest accuracy, in terms of the number of parameters, FLOPS and model size, the proposed approach is significantly better especially compared to Inception, Resnet18, and Alexnet approaches. Please also note that for relatively deeper models such as Inception, images of size $300\times 300\times 3$ should be used, since $64\times 64\times 3$ images vanish during the process of the forward pass due to the pooling and padding operations.

We end this section with a summary of the performance of the proposed model in terms of accuracy. The results of the evaluation carried out show that with an accuracy of about 98.28%, the proposed model is superior to studies such as References [21, 30, 43, 56], which report accuracies of 94.39%, 95.45%, 95%, and 86%, respectively. The study presented in Reference [36] performs slightly better than the proposed architecture with an accuracy of 99%; however, it is primarily based on GoogLeNet, which is a 22-layers convolutional neural network. The well-known architectures Mnasnet, Mobilenet, Sufflenet, Inception, Resnet, and Alexnet provide accuracies of 64.05%, 97.58%, 98.81%, 99.72%, 99.00%, and 99.00%, respectively. In terms of accuracy, the proposed approach performs close to the approaches with the highest accuracies.