# COMPARISON OF CLASSIFICATION AND DETECTION OF FOREST FIRES USING MOBILENETV2, SQUEEZENET AND FIRENET

**OKAN ARSLAN**

**September 18, 2023**

# 1 Introduction

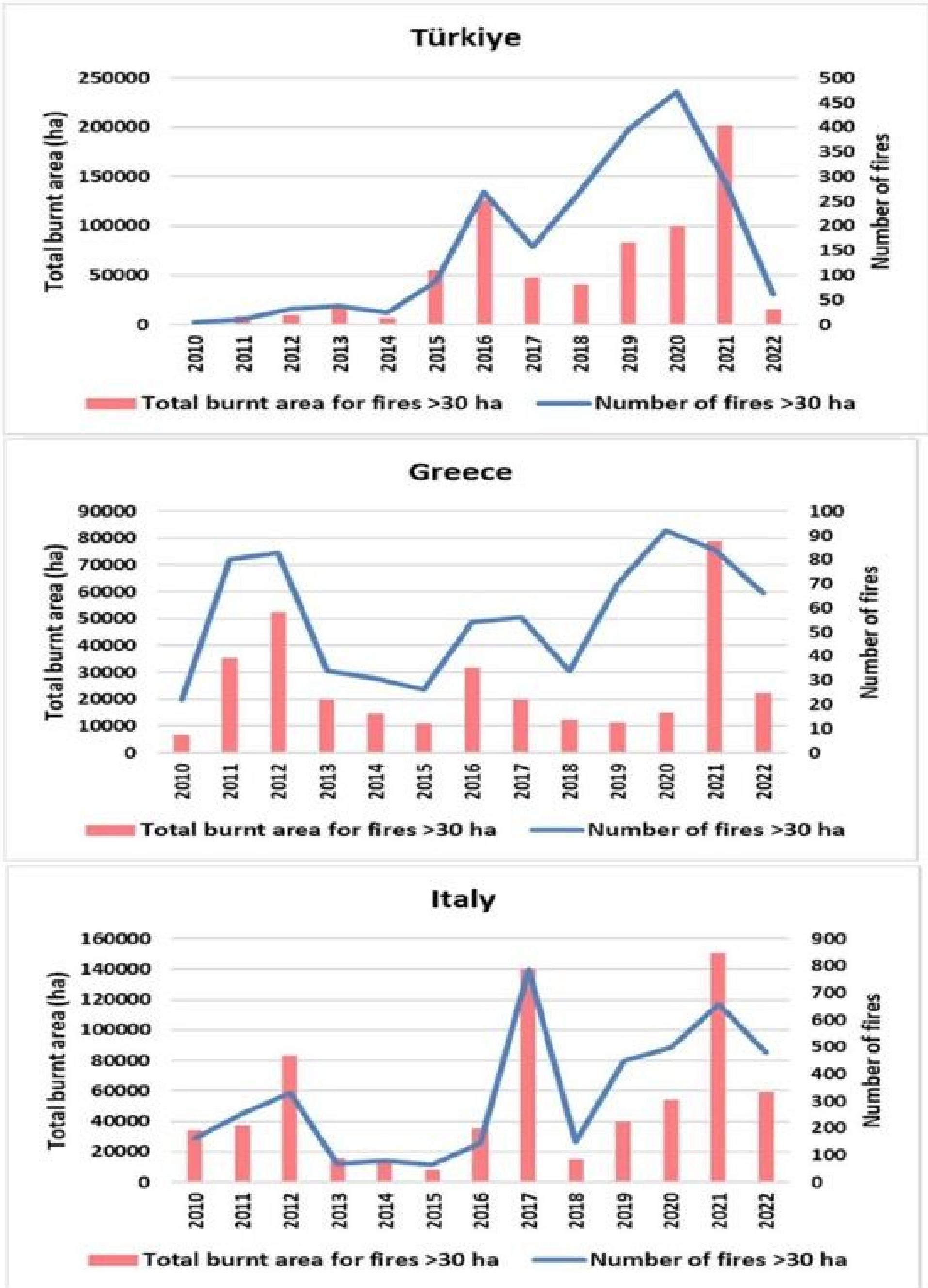


**Figure 1:** Graph of reviews by [1] and In the graph, the values of Total burnt area for fires >30ha and Number of fires >30ha are provided for Turkey, Greece, and Italy. It is notable that these values have been increasing since 2018.

Fires pose a significant threat every year, especially during the summer months, in warm climate countries like those in the Mediterranean region. These fires often spread extensively before they are even noticed or brought under control.

The motivation behind addressing this problem lies in the need for early response to the early detection of forest fires. By detecting fires promptly and taking immediate action, we can mitigate their destructive impact, preserve ecological balance, and prevent further exacerbation of climate change. Utilizing advanced technologies plays a crucial role in environmental protection and fire management.

In this report, we aim to explore and compare different technologies used for detecting and combating forest fires. By evaluating and analyzing various approaches, such as remote sensing, image analysis, and data-driven models, we can gain insights into their effectiveness and suitability in addressing this critical issue.

Furthermore, as part of this report, we will refer to the Advance Report on Forest Fires in Europe, Middle East, and North Africa 2022 for relevant information and data. Figure 1 illustrates the annual mapped burnt area of fires with a size equal to or larger than 30 hectares in Turkey, Greece, and Italy, providing valuable insight into the severity and extent of the fire problem in these countries.

Through this report, our aim is to highlight the importance of addressing forest fires, explore advancements in technology, and propose effective strategies for early detection and mitigation. By doing so, we hope to contribute to the preservation of natural resources, protection of ecosystems, and enhancement of fire management practices.

## 2 Dataset Description

**FIRE Dataset**

The dataset used in this project was created by our team during the NASA Space Apps Challenge in 2018. The primary objective of the dataset was to develop a model capable of accurately recognizing images containing fire. For further information regarding the context or details about the challenge, please refer to our team page.

The dataset consists of two main folders,fire images and non-fire images. The fire images folder contains a total of 755 outdoor fire images, some of which depict scenes with heavy smoke. On the other hand, the non-fire images folder consists of 244 nature images, encompassing various subjects such as forests, trees, grass, rivers, people, foggy forests, lakes, animals, roads, and waterfalls.

It is important to note that the dataset follows a binary classification problem, aiming to distinguish between images that contain fire and regular images. However, there exists an inherent class imbalance within the data set, meaning that the two classes (folders) do not possess an equal number of samples. To address this imbalance, it is recommended to create a validation set that includes an equal number of images from both the fire and non-fire classes. For instance, allocating 40 images per class would ensure an equitable representation in the validation set. Here, visual representation of the data set's structure and content refer to Figure 2.

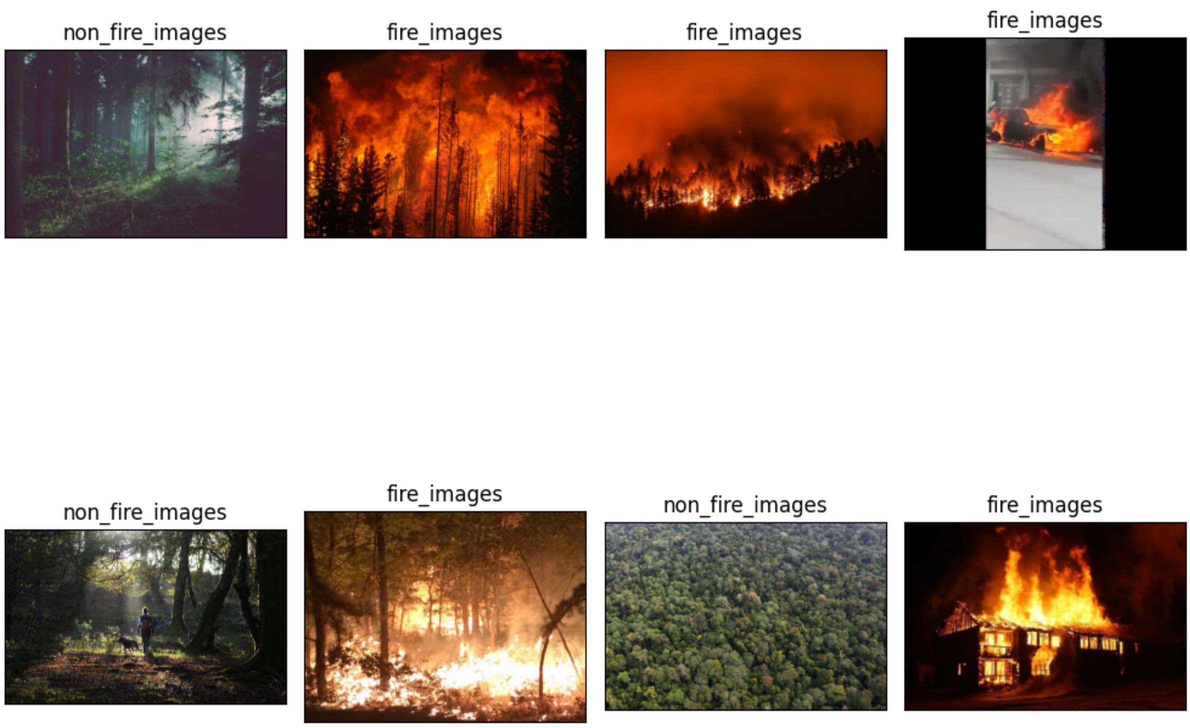


**Figure 2:** Data structure and content representation. The Examples of "fire images" and The examples of "non fire images" can be seen.

# 3 Novelty

Deep learning approaches, including convolutional neural networks (CNN) such as SqueezeNet and MobileNetV2, have emerged as prominent methods for wildland fire classification, detection, and segmentation. These models have demonstrated remarkable performance and have been widely adopted in the field.

SqueezeNet, introduced by Iandola et al. [2], is a compact CNN architecture that achieves high accuracy while minimizing model size. It employs fire modules, consisting of 1x1 and

3x3 convolutions, to capture complex spatial patterns and features. By reducing the number of parameters, SqueezeNet offers an efficient solution for wildland fire analysis. MobileNetV2, proposed by Sandler et al. [3] , is specifically designed for mobile and embedded vision applications. It utilizes depthwise separable convolutions, linear bottlenecks, and inverted residuals to achieve lightweight models without sacrificing performance. This makes MobileNetV2 an attractive option for resource-constrained environments.

Furthermore, it is possible to create custom network architectures by adjusting the layers and parameters based on the specific requirements of wildland fire tasks. Researchers can experiment with different combinations of convolutional, pooling, and fully connected layers, as well as incorporate regularization techniques like dropout and batch normalization. This flexibility allows for the development of tailored networks that address the unique characteristics and challenges of wildland fire imagery, potentially improving overall performance and accuracy.

The novelty of these deep learning approaches lies in their ability to leverage large-scale datasets and learn intricate representations of wildland fires. By automatically learning and adapting to the complexities and variabilities present in fire images, these models outperform traditional machine learning methods and significantly enhance wildland fire detection, segmentation, and classification tasks.

To the best of our knowledge, there is a limited number of published studies that specifically address the implementation of deep learning models, such as SqueezeNet and MobileNetV2, in the context of wildland fire analysis. Therefore, this review aims to bridge this gap by providing an up-to-date and comprehensive analysis of these vision methods and their performances in the wildland fire domain. The challenges and open research directions in this field are also discussed, paving the way for future studies to develop more accurate and robust models for wildland fire remote sensing.

# 4 Model Description

This section encompasses the details of three utilized models: MobileNetV2, SqueezeNet, and FireNet.

## 4.1 MobileNetV2

MobileNetV2 is an efficient, feature-rich convolutional neural network (CNN) architecture well-suited for mobile and embedded vision applications due to its low computational complexity. The architecture is based on an inverted residual structure where the input and output of the residual block are thin bottleneck layers, contrary to traditional residual models. This structure results in improved model efficiency.

In this study, a pre-trained MobileNetV2 model with weights from 'Imagenet' is employed, configured with input shape as 224x224x3 and excluding the top or final fully connected layer. The model is initially kept non-trainable to retain the learned features, following which a set of dense and dropout layers are added for further fine-tuning and to prevent overfitting.

The final dense layer consists of 2 neurons, representing the two classes in our classification problem. The model is compiled using Adam optimizer with a learning rate of 0.0001, categorical cross entropy as the loss function, and accuracy as the metric. Figure 3 shows the architecture of MobileNetv2.

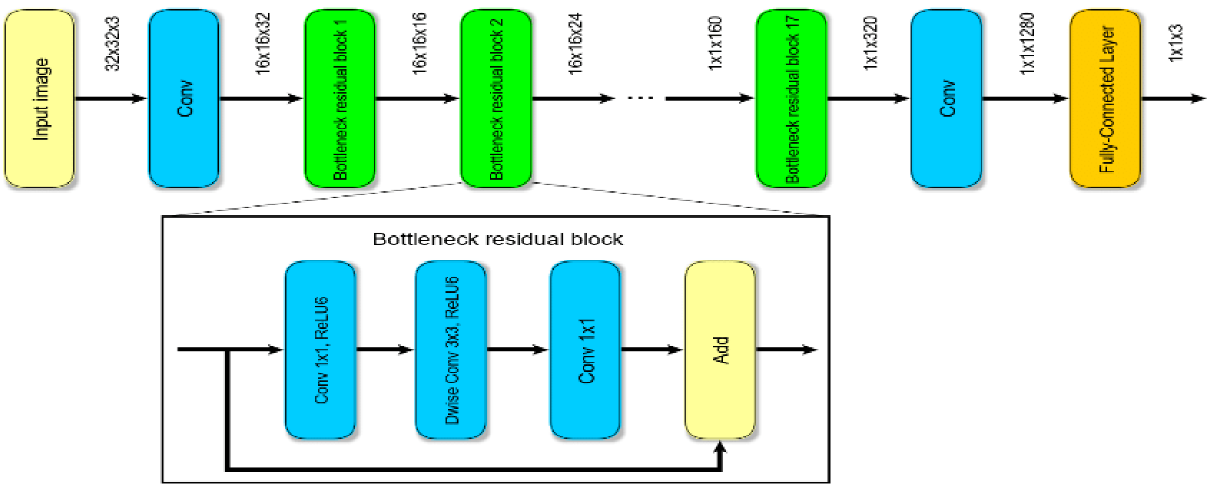


**Figure 3:** The architecture of the MobileNetV2 network.
[4]

## 4.2 SqueezeNet

SqueezeNet is a deep learning architecture that consists of multiple layers designed for image classification tasks. It achieves a good balance between model size and accuracy by using a combination of 1x1 and 3x3 convolutions.

The main components of SqueezeNet include:

- Convolutional Layers: These layers extract features from the input images using filters that slide over the image and perform convolutions.
- Fire Modules: These modules consist of a squeeze layer (1x1 convolutions) followed by expand layers (1x1 and 3x3 convolutions). They help capture both spatial and channel-wise information in an efficient manner.
- Max Pooling Layers: These layers reduce the spatial dimensions of the feature maps while preserving the most important information.
- Dropout: This regularization technique randomly drops some units during training, preventing overfitting and improving generalization.
- Fully Connected Layers: The final layers of the network aggregate the extracted features and make predictions based on the learned representations.

By leveraging these components, SqueezeNet achieves high accuracy with fewer parameters compared to other architectures. It is particularly suitable for resource-constrained environments where model size and computational efficiency are crucial factors. In this study, SqueezeNet was analyzed in 2 different ways. In the first one, it was used with PyTorch, which is its own usage. In the second part, it was processed by adjusting the SqueezeNet layers with tensorflow. This was only tried for a comparison within itself. The architecture of the SqueezeNet network is shown in Figure 4.

### 4.2.1 SqueezeNet with PyTorch

The SqueezeNet model, which has been pre-trained on a large dataset, is utilized in this implementation. To load the model, the PyTorch hub library is used, and the original fully connected layer at the end of the network is replaced with a new one. The new fully connected layer is designed with two neurons to align with the specific classification problem, where there are two classes to predict.

After loading the model, it is transferred to the available computing device, which could be either a GPU (Graphics Processing Unit) or a CPU (Central Processing Unit). This step

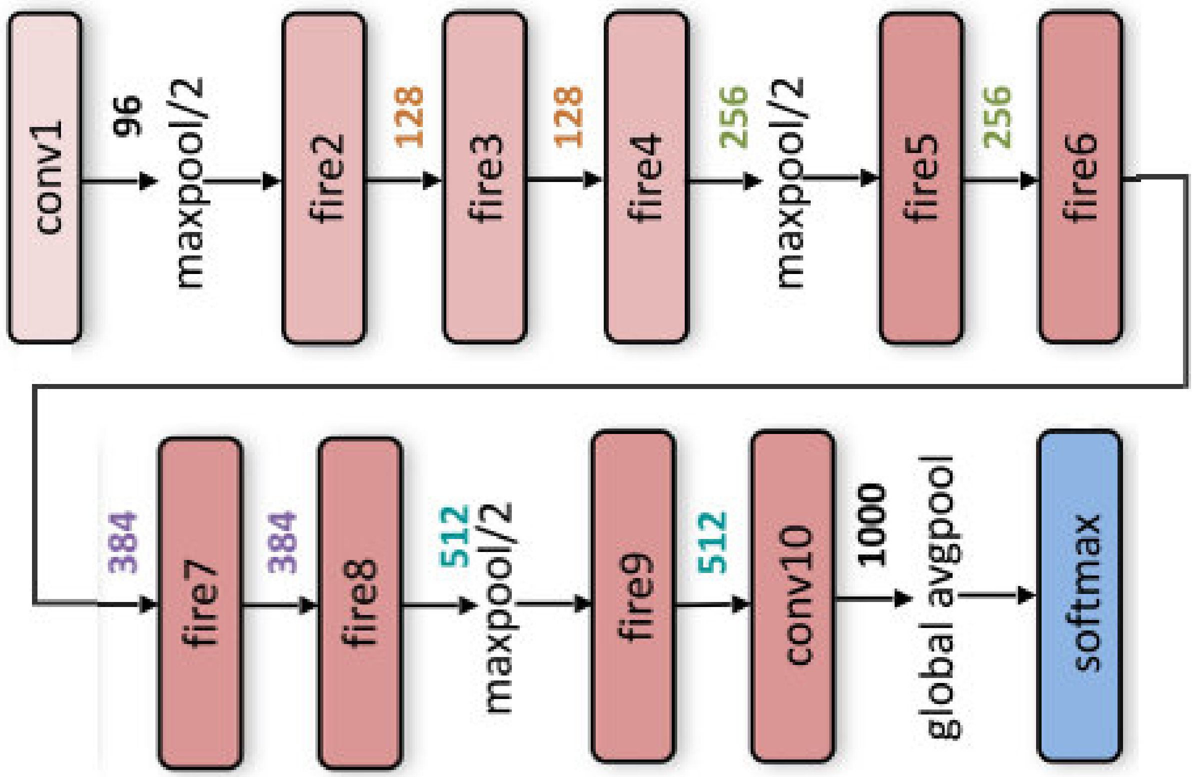


**Figure 4:** The architecture of the SqueezeNet network.
[5]

ensures that the model is ready to perform computations on the chosen device, leveraging its computational power for efficient and effective inference.

#### 4.2.2 SqueezeNet with TensorFlow

To implement SqueezeNet with TensorFlow, we define the "fire module" first. Each fire module consists of a squeeze convolution layer (1x1 filters), followed by an expand layer that has a mix of 1x1 and 3x3 convolution filters. The squeeze and expand layers introduce a mechanism for channel-wise feature recalibration, contributing to SqueezeNet's efficiency.

The SqueezeNet model is built by stacking these fire modules, with max-pooling layers for downsampling at specific intervals, and a final convolution and average pooling layer for classification. Dropout is applied for regularization. The model is then compiled with the Adam optimizer, categorical cross entropy loss, and accuracy as the metric.

## 4.3 FireNet

FireNet is a convolutional neural network (CNN) architecture. The use of FireNet in this study inspired by the work titled "FireNet: A Specialized Lightweight Fire and Smoke Detection Model for Real-Time IoT Applications."[6] The input size of FireNet is 224x224x3, representing the width, height, and color channels of the input images.

The network consists of several convolutional layers, each followed by a Rectified Linear Unit (ReLU) activation function. These convolutional layers extract and capture important features from the input images. Max-pooling layers are incorporated throughout the network to downsample and reduce the spatial dimensions, enabling the model to learn hierarchical representations effectively.

After the convolutional base, the feature maps are flattened, resulting in a 1D vector representation. This vector is then fed into a series of dense layers for classification. Dropout layers are inserted between the dense layers with a dropout rate of 0.3 to regularize the model and prevent overfitting.

The final dense layer of FireNet consists of 2 neurons, corresponding to the two classes of interest (e.g., fire and non-fire). The softmax activation function is applied to the output layer, enabling the model to generate class probabilities.

The architecture and summary of FireNet can be found in Table 1. It should be noted that FireNet's design is optimized for lightweight and real-time inferencing in IoT environments. The model is compiled using the Adam optimizer with a learning rate of 0.0001 and categorical cross-entropy as the loss function, ensuring efficient training and accurate classification results.

**Table 1:** Model Architecture Summary

| Layer (type) | Output Shape | Param # |
|---|---|---|
| conv2d | (None, 54, 54, 96) | 34,944 |
| max_pooling2d | (None, 26, 26, 96) | 0 |
| conv2d_1 | (None, 22, 22, 256) | 614,656 |
| max_pooling2d_1 | (None, 10, 10, 256) | 0 |
| conv2d_2 | (None, 6, 6, 384) | 2,457,984 |
| max_pooling2d_2 | (None, 2, 2, 384) | 0 |
| flatten | (None, 1536) | 0 |
| dropout | (None, 1536) | 0 |
| dense | (None, 2048) | 3,147,776 |
| ... | ... | ... |
| Total params | 8,355,586 | |
| Trainable params | 8,355,586 | |
| Non-trainable params | 0 | |

# 5 Experimental Results

The results obtained in this study will be analyzed under this heading. The data described so far were used to train the models after the necessary preparations were made. Separately, these networks were subjected to testing. The results can be examined in Table 2 . If we compare the performance of each model, we see that MobileNetv2 has the highest overall accuracy, precision and F1-Score values. However, FireNet's ability to detect non-fire images (recall) is lower. This may perhaps indicate that the model tends to misclassify non-fire images as fire.

**Table 2:** Model Performance Comparisons

| | Model | Precision | Recall | F1-Score | Accuracy |
|---|---|---|---|---|---|
| Fire Images | FireNet | 0.960526 | 0.986486 | 0.973333 | 0.960000 |
| | MobileNetv2 | 0.980132 | 1.000000 | 0.989967 | 0.985000 |
| | SqueezeNet | 0.98 | 0.98 | 0.98 | 0.98 |
| Non-Fire Images | FireNet | 0.958333 | 0.884615 | 0.920000 | 0.960000 |
| | MobileNetv2 | 1.000000 | 0.942308 | 0.970297 | 0.985000 |

Talking about the SqueezeNet model, when trained and tested with Pytorch, although the result was close to the MobileNetv2 model and better than the FireNet model, the model could not be successfully deployed using Tensorflow. This could be due to a lack of a compatible version or a software/hardware issue that does not support a specific feature. This directly affects the performance of the model because we are not able to fully utilize it. In Table 3 we can observe the results of training SqueezeNet with TensorFlow extracted from the overall benchmark.

**Table 3:** SqueezeNet with Tensorflow Performance Measures

| | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| Fire Images | 0.758170 | 0.783784 | 0.770764 | 148.00 |
| Non-Fire Images | 0.319149 | 0.288462 | 0.303030 | 52.00 |
| Accuracy | 0.655000 | | | |
| Macro Avg | 0.538659 | 0.536123 | 0.536897 | 200.00 |
| Weighted Avg | 0.644024 | 0.655000 | 0.649153 | 200.00 |

# 6 Conclusion and Ideas

Early detection and monitoring of forest fires is one of the most effective ways to contain them and prevent their spread. However, it is difficult to monitor such events, which often occur in hard-to-reach areas. This is where AI technologies, and deep learning models in particular, can come into play. Aerial vehicles, such as drones, are ideal tools in this regard because they can often reach such difficult terrain and can quickly scan large areas. Deep learning models such as SqueezeNet, MobileNetv2 and FireNet could potentially be used in this case.

SqueezeNet and MobileNetv2 are both known as lightweight and efficient models. Although they use fewer parameters, they can often perform similarly to larger and more complex models. This increases the usability of such models in flying devices such as drones, which have limited processing power. Such devices may not usually have a built-in GPU or a high-performance processor, so lighter models offer a big advantage here.

FireNet can be used in a similar way but will consume slightly more resources. This means that a lightweight system such as the Raspberry Pi or Jetson Nano may be advantageous. The Raspberry Pi is an easily available mini-computer that is supported by a large community and offers low energy consumption and reasonable processing power. FireNet can run on such a system and utilize its fire detection capabilities. On the other hand, Nvidia's Jetson Nano platform could offer a more powerful option for drones or other portable devices. Jetson Nano offers more processing power and especially GPU acceleration capabilities. This could be ideal for models that consume slightly more resources, such as FireNet. Also, the size and weight of the Jetson Nano is such that it is still suitable for drones or other portable devices. Ultimately, the model and hardware chosen will depend on the use case and available resources. SqueezeNet and MobileNetv2 can operate on lightweight, energy-efficient hardware, while FireNet may require a more powerful platform, such as the Jetson Nano. Such technological combinations can play an important role in the early detection and monitoring of forest fires and can be a major step forward in minimizing the impact of such disasters.

## Acknowledgment

- We would like to thank the Institute of Advanced Research in Artificial Intelligence (IARAI) GmbH, Austria, for making this dataset publicly available.
- The dataset used in this project, named the Fire Dataset, was created by Ahmed Gamaleldin, Ahmed Atef, Heba Saker, and Ahmed Shaheen. It is publicly available at the following address: [https://www.kaggle.com/datasets/phylake1337/fire-dataset]
- I would like to express our gratitude to the creators for making this dataset accessible, enabling researchers and practitioners to explore and contribute to the field of fire image recognition. I would also like to thank all the participants who have developed projects on the dataset I was inspired by.


## References


[1] SAN-MIGUEL-AYANZ Jesus; DURRANT Tracy; BOCA Roberto; MAIANTI Pieralberto; LIBERTA' Giorgio; OOM Duarte; BRANCO Alfredo; DE RIGO Daniele; FERRARI Davide; ROGLIA Elena; SCIONTI Nicola. Annual mapped burnt area of fires >=30ha in turkey, greece and italy. *JRC Technical Report:Advance Report on Forest Fires in Europe, Middle East and North Africa*, pages 24,26,37, 2023.

[2] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and <0.5 mb model size. *arXiv preprint arXiv:1602.07360*, 2016.

[3] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.

[4] Lyazzat Ilipbayeva Eric T. Matson Ulzhalgas Seidaliyeva, Daryn Akhmetov. Real-time and accurate drone detection in a video with a static background. Scientific Figure on ResearchGate, 2020.

[5] Xuenan Cui Van Huan Nguyen Thi-Hai-Binh Nguyen, Eunsoo Park. fpadnet: Small and efficient convolutional neural network for presentation attack detection. Scientific Figure on ResearchGate, 2018.

[6] Akshay Varshney Mohammad Samar Ansari Rishabh Sharma Arpit Jadon, Mohd. Omama. Firenet: A specialized lightweight fire smoke detection model for real-time iot applications. 2019.

[7] Rafik Ghali and Moulay A Akhloufi. Deep learning approaches for wildland fires remote sensing: Classification, detection, and segmentation. *Remote Sensing*, 15(7):1821, 2023.

[8] Mesut Toğaçar, Burhan Ergen, and Zafer Cömert. Covid-19 detection using deep learning models to exploit social mimic optimization and structured chest x-ray images using fuzzy color and stacking approaches. *Computers in Biology and Medicine*, 121:103805, 2020.