IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 4, 2022, accepted February 8, 2022, date of publication February 15, 2022, date of current version March 7, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3151660

Convolution Optimization in Fire Classification

ANGEL AYALA[1], BRUNO JOSÉ TORRES FERNANDES[1], (Member, IEEE), FRANCISCO CRUZ[2,3], DAVID MACÊDO[4,5], (Graduate Student Member, IEEE), AND CLEBER ZANCHETTIN[4,6], (Member, IEEE)

[1]Escola Politécnica de Pernambuco, Universidade de Pernambuco, Recife 50720-001, Brazil
[2]School of Information Technology, Deakin University, Geelong, VI 3225, Australia
[3]Escuela de Ingeniería, Universidad Central de Chile, Santiago 8330015, Chile
[4]Centro de Informática, Universidade Federal de Pernambuco, Recife 50720-001, Brazil
[5]Montreal Institute for Learning Algorithms, University of Montreal, Montreal, QC H3T 1J4, Canada
[6]Department of Chemical and Biological Engineering, Northwestern University, Evanston, IL 60208, USA

Corresponding author: Angel Ayala (aaam@ecomp.poli.br)

This work was supported in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior—Brazil (CAPES)—Finance Code 001, in part by the Fundação de Amparo a Ciência e Tecnologia do Estado de Pernambuco (FACEPE), and in part by the Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq)—Brazilian Research Agencies.

ABSTRACT Early alert fire and smoke detection systems are crucial for daily and security management decision-making. Recent literature approaches are based on Deep Learning (DL) models. Efficient models are required for hardware-constrained systems, such as mobile devices, embedded systems, and robotics achieving high performance at low power consumption. For this research, we designed a novel specific-purpose model for fire and smoke recognition using still images and the study of state-of-the-art convolution techniques to improve the trade-off between accuracy and complexity. In this regard, the literature suggests that the inverted residual block, the depthwise and octave convolution techniques, reduces the model's size and computation requirements working well by themselves. In this work, we propose the KutralNext architecture, an efficient model for single- and multi-label fire and smoke recognition tasks. Additionally, a more efficient architecture KutralNext+, demonstrates that those convolution techniques achieve a better trade-off combined, reaching an 84.36% average test accuracy in FireNet, FiSmo, and FiSmoA fire datasets. The KutralSmoke and FiSmo fire and smoke datasets attained an 81.53% average test accuracy. Furthermore, a previous fire and smoke recognition model considered, FireDetection, KutralNext uses 59% fewer parameters, and KutralNext+ requires 97% fewer flops and is 4x faster.

INDEX TERMS Efficient approach, fires, image classification, convolutional neural networks.

I. INTRODUCTION

The presence of unrestrained fire in any environment is a disaster that causes economic and ecological damage, endangering people's lives [1]–[4]. Highly fire-hazard industries need risk assessment tools to reduce the fire occurrence probability using preventive actions [5], such as storage protocols for different combustibles. However, when a blaze appears, reactive tools are needed [6] at the early stages of the combustion in the case of preventive action failure. Early warning devices are essential to managing fire disasters, reducing the damage. Nevertheless, traditional sensor-based systems are not quick enough to sense the fire [7], [8]. Image-based fire recognition methods are a new and promising approach [9], considering the outstanding results on image processing applications of deep learning (DL) methods [10].

Deep learning has been applied in many tasks using images, such as classification [11], object detection [12], and segmentation [13]. Current DL methods are successful in computer vision tasks since it is the most approximated way to model a visual data input into useful information mathematically [14]. The input image is processed through each one of the model's layers. The deeper the convolutional neural network is, the richer low-dimensional features it can obtain.

Moreover, current DL methods have been proven to surpass the human eye performance in recognizing objects [15]. Many of these DL models achieved state-of-the-art performance in the ImageNet ILSVRC dataset, improved to obtain even higher efficacy. For example, in 2015, GoogLeNet [16] achieved 74.8% top-1 accuracy with 6.8M parameters.

The associate editor coordinating the review of this manuscript and approving it for publication was Kuo-Ching Ying.

More recently, GPipe [17] achieved 84.3%, using 557M parameters. Nevertheless, Tan *et al.* [18] demonstrated that deeper networks do not always perform better or more efficiently. The authors studied how the width, resolution, and depth are related when scaling up a model in their research.

Many neural networks developed from scratch can be treated as a heuristic task, in which the purpose is to surpass the state-of-the-art performance. However, as pointed out by Tan *et al.* [18], other aspects must be considered during the architecture's design. These characteristics include width, resolution, and depth as correlated values. This research addresses the feasibility of mixture DL techniques to attain the lowest model complexity with the highest accuracy. In recent studies, the inverted residual block [19] has been proven to be an efficient convolution block architecture [20], [21]. Additionally, the octave convolution [22] with their high- and low-frequency signal processing method has shown to be a suitable replacement for the vanilla convolution [23]–[25].

Ayala *et al.* proposed a lightweight approach for fire recognition using two benchmark datasets to test generalization under the same and different data distribution. The best accuracy of their model was under a balanced class partitioning, demonstrating that KutralNet is a suitable low computational cost fire classifier. This research proposes an extension of KutralNet architecture to prove the feasibility and efficiency in combining the inverted residual block with the depthwise and octave convolution, named KutralNext. First, a baseline model is defined with vanilla convolutions as a reference to a more efficient model, replacing some parts with combined cost-effective DL techniques. Hereof, KutralNext also proposes recognizing fire and smoke as a multi-label approach giving separated inferences to assess the fire dimensions or intensity. Additionally, to improve the results and solve the lack of balanced datasets, the ImageNet dataset is used for pretraining the models and the class balanced loss function to deal with unbalanced instances. Moreover, a newly compiled correctly labeled dataset is proposed as a new multi-label dataset to recognize fire and smoke in still images. We compared the proposed models with state-of-the-art fire, and smoke recognition approaches over the FireNet [26], FiSmo [27], and FireSmoke [28] datasets. The KutralNext proposes recognizing fire and smoke presence in an image using the deep learning methodology and a newly compiled correctly labeled dataset. Our model's primary focus is to reduce the complexity of processing an image by a model capable of running in embedded resource-constrained platforms, like closed-circuit TV systems, mobile devices, and robotic systems, and warning about the existence of a fire emergency scenario.

A. RELATED WORKS

The first approaches to fire recognition in computer vision were addressed using RGB color space [29], spectral color [30], texture recognition [31], and spatio-temporal treatment [32]. Recent work [33] has focused on optimizing the response time of a computer vision system by reducing the number of frames processed through a convolutional neural network (CNN). A motion detection stage skips the unprocessed frames based on a background model. Spatio-temporal information analysis is helpful to discriminate fixed pixel values from fluctuating pixel values, this in a determined number of a video frames sequence, identifying as motion when a pixels transition from a fixed to another different value is encountered. The most recent methods follow this line using deep neural networks with CNNs.

Many DL implementations were built on previously trained models [34]–[36], such as ResNet and its variations [37]. Others were designed for this specific purpose [26], [38] to recognize fire presence in images. More recent studies [39], [40] focused on identifying the presence of fire and smoke images, considering three outputs: one for fire, one for smoke, and another for both, addressing the problem as a single-label classification task.

A lightweight approach to fire recognition was addressed by Jadon *et al.* [26] where the authors proposed a fire recognition model with three-convolutional layers to process a 64×64 RGB input image, trained and tested against the author's dataset, obtaining a 93.91% in test accuracy. The proposal used the CNN as part of an embedded IoT fire alarm system from visual input. That is why the authors designed a lightweight model to run at a high frame rate with low resources.

Another lightweight inspired approach was achieved by Ayala *et al.* [38] where the authors combine the Octave convolution with the ResNet architecture presenting a few residual blocks with the shortcut connection. The authors compared the model with four datasets previously used in this task, obtaining an 87.44% average of validation accuracy. The dataset included FireNet [26] and FiSmo [27] original datasets.

Gotthans *et al.* [39] proposed the Fire Detection model to fire and smoke recognition trained with two datasets to compare it against AlexNet [11] and SqueezeNet [41]. The model received an input image of 224×244 pixels with RGB channels, normalized with average values of (0.485, 0.456, 0.406) and standard deviation of (0.229, 0.224, 0.225) for each channel. The training was performed during 100 epochs, with a batch size of 20 and a learning rate of 0.001. They proposed a lightweight model capable of recognizing fire and smoke in still images. The Fire Detection model reduced by 27% the execution time compared to AlexNet, achieving 1% less accuracy. However, only the validation accuracy of the experiments was presented. Additionally, they executed the model in the Jatson Nano platform with similar results during training.

Oh *et al.* [40] used the EfficientNet-B0 [18] model to recognize a fire emergency from images. Using an automated algorithm, they elaborated a new dataset from various image search engines to collect cloud, snow, rural, fire, wave, and waterfall labeled images. Next, they made a manual cleanup obtaining a total of 14,741 images. In this case, the

fire-labeled images contained an open-air environment with the presence of fire, smoke, or both. The model was trained using Focal Loss [42] to deal with dataset class imbalance, in addition to random data augmentation, over 90 epochs, and a batch size of 256. A pre-trained version of the model with ILSVRC2012 was also used with fine-tuning. The proposal achieved a high test accuracy of 99.05% with their datasets.

Much current development using the DL method combined with fine-tuning or transfer-learning techniques solves the lack of training data. The only issue is the network's size and complexity constraint. Most proposals focus on solving and surpassing the current state-of-the-art accuracy, leaving out resource constraints and requirements.

II. MODEL EFFICIENCY TECHNIQUES BACKGROUND

With the success of deep convolutional neural networks, efficient techniques appeared with newly proposed models. The first known technique used is the residual connection [37]. It can be formally defined as $\mathcal{H}(x) = \mathcal{F}(x) + x$ where x is the input signal or the identity connection, and $\mathcal{F}(x)$ is the convoluted input signal. This strategy reduces the overfitting during the training of deeper architectures, improving the gradient's propagation across multiple layers requiring almost the same number of operations.

The second widely used technique is depthwise separable convolution. It separates the convolution in a channel-wise spatial correlation mapping, followed by a cross-channel mapping with a 1×1 convolution called pointwise convolution. Chollet *et al.* [10] proposed the Xception model, where the authors used depthwise separable convolutions and residual connections in the architecture. Depthwise convolution was first presented by Sifre *et al.* [43]. Chollet [10] proved its efficiency in-depth with its Xception model, which uses depthwise separable convolution and residual connections in almost all the architecture.

If the computational cost of a vanilla convolution is given by

$$\mathcal{C}_v = D_k * D_k * M * N * D_f * D_f, \tag{1}$$

where D_k is the kernel size, assumed square, M is the number of input channels, N is the number of output channels, and D_f is the feature map size. According to the definition, the computational cost of depthwise convolution is given by

$$\begin{aligned} \mathcal{C}_{dw} &= D_k * D_k * M * D_f * D_f \\ &= \frac{\mathcal{C}_v}{N}. \end{aligned} \tag{2}$$

On the one hand, depthwise convolution is more efficient than vanilla convolution because it breaks the relationship between output channels, as presented in (2). On the other hand, the pointwise convolution computational cost is given by

$$\begin{aligned} \mathcal{C}_{pw} &= M * N * D_f * D_f \\ &= \frac{\mathcal{C}_v}{D_k^2}, \end{aligned} \tag{3}$$

breaking the relationship between the kernel size, as presented in (3).

A third most commonly used technique is the inverted residual block [19]. It is composed of depthwise separable convolution and residual connections. The peculiarity of this convolutional block is the presence of the shortcut in the bottleneck between the layers with a low number of channels, as can be observed in Figure 1(a). Additionally, it presents an expansion layer that increases the number of channels processed between the bottleneck using a depthwise convolution, before and after pointwise convolution. This convolutional block presents a computational cost that depends on an expansion rate t:

$$\begin{aligned} \mathcal{C}_{irb} &= D_f * D_f * M * t(2N + D_k * D_k), \\ &= t(2 * \mathcal{C}_{pw} + \mathcal{C}_{dw}) \end{aligned} \tag{4}$$

in comparison with vanilla convolution, this block obtain a reduction of

$$\mathcal{W}_{irb} = \frac{\mathcal{C}_{irb}}{\mathcal{C}_v} = \frac{2}{D_k^2} + \frac{1}{t * N}. \tag{5}$$

Another new technique for efficient model design is octave convolution [22]. It decomposes the input signal in a high-spatial frequency to describe the rapidly changing details and in a low-spatial frequency to describe the smoothly changing structure. The authors have demonstrated that using the octave convolution in popular DL models like ResNet [37] consistently improves the results, reducing the flops and model size. Formally, let X be the input image $\in \mathbb{R}^{M * D_f * D_f}$, where D_f is the spatial dimension considered squared, and M the number of channels. X is factorized into $X = \{X^H, X^L\}$, considering $X^H \in \mathbb{R}^{(1-\alpha)M * D_f * D_f}$ the high-frequency feature maps of fine details, and $X^L \in \mathbb{R}^{\alpha M * \frac{D_f}{2} * \frac{D_f}{2}}$ the low-frequency feature maps of general characteristics. Here $\alpha \in [0, 1]$ is a hyper-parameter denoting the ratio of channels allocated in the low-frequency part. In this regard, the computational cost consists of two components given by

$$\mathcal{C}_o = \mathcal{C}_v * (1 - \alpha) + \mathcal{C}_v * \frac{\alpha}{4}, \tag{6}$$

obtaining a reduction dependent of α

$$\mathcal{W}_o = \frac{\mathcal{C}_o}{\mathcal{C}_v} = (1 - \alpha) + \frac{\alpha}{4}. \tag{7}$$

Each side of the equation's addition represents the convolutional cost for the high- and low-frequency. After processing the signal, each frequency feature map's information is exchanged between them, as shown in Figure 1(b).

Considering that (2) reduces (1) in terms of N and (6) weight (1) into two terms, we can assume that the inverted residual block with octave convolution cost is given by:

$$\mathcal{C}_{irb8} = t(2 * \frac{\mathcal{C}_o}{D_k^2} + \frac{\mathcal{C}_o}{N}), \tag{8}$$

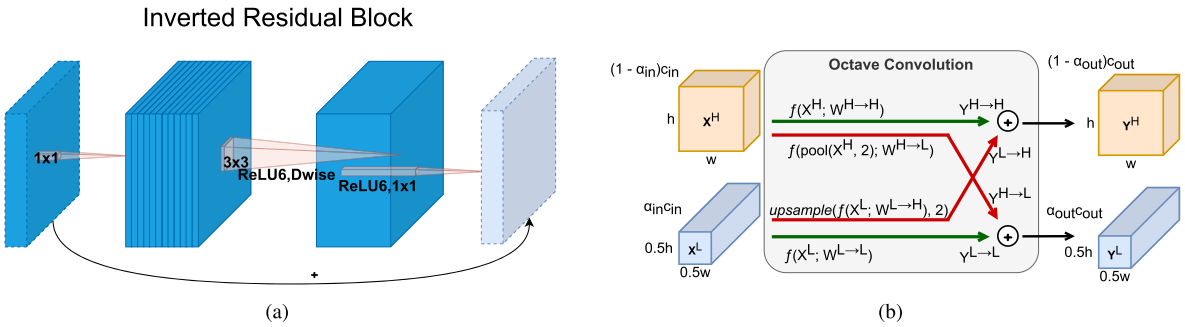


FIGURE 1. Main DL techniques used in this research: (a) The inverted residual block. Diagonally hatched layers do not use non-linearities. The thickness of each block is used to indicate its relative number of channels. The inverted residuals connect the bottlenecks. Adapted from [19]. (b) Detailed design of the octave convolution. Green arrows correspond to information updates, while red arrows facilitate information exchange between the two frequencies. Adapted from [22].

obtaining a reduction dependent of (7)

$$\mathcal{W}_{irb8} = \frac{\mathcal{C}_{irb8}}{\mathcal{C}_{irb}} = \frac{\mathcal{C}_o}{\mathcal{C}_v}. \tag{9}$$

III. KUTRALNEXT

This work proposes KutralNext, a model to fire and smoke image recognition under a multi-label approach, based on the KutralNet architectures to process 84 × 84 pixels RGB images. The KutralNext baseline model uses vanilla convolution on its stack of layers with two units as exit, one for each fire and smoke case, to work under a low computational cost. The more efficient version, named KutralNext+, is built on the inverted residual block [19], depthwise convolution [43], and octave convolution [22], modifying the baseline blocks with those techniques. As the literature suggests, pretrained models with the ImageNet dataset increase the number of filters learned by a model. Therefore, the baseline and efficient models are pre-trained using the ILSVRC2012 dataset and then fine-tuned with the fire and smoke datasets. Furthermore, it is uncommon to find balanced datasets between its labels, and therefore, it is included the class balanced loss to solve these disproportional number of instances. More details about the architecture of each model, ILSVRC2012 dataset, loss function, and multi-label approach are described in the following subsections.

A. BASELINE ARCHITECTURE

The baseline model comprises three kinds of convolutional blocks, named KutralBlockN (KBN), where N corresponds to the number of output channels, KutralBlockP (KBP), and KutralBlockO (KBO). KBN block was built up with a convolution layer with N channels as output, a batch-normalization layer, a LeakyReLU activation, and a max-pooling layer to size down the output. Next, the KBP block possesses two convolution layers and a batch-normalization layer. Finally, the KBO block comprises a LeakyReLU activation, a global average pooling layer, and a fully-connected layer. More details for each block are present in Figure 2(a).

As illustrated in Figure 2(b), the model consists of three KBN blocks, one KBP block followed by a shortcut of max-pooling, and batch-normalization layers. This KBN is the block that processes the signal from the KB64 block and, finally, a KBO block. The KBO block's fully-connected layer presents a variation for a single- and multi-label approach specified in section III-D. This approach has proved that few layers can acquire enough features for a fire classification task to optimize the inference time [26]. Additionally, using a shortcut and batch-normalization layer avoids overfitting the model [37]. We have chosen LeakyReLU as a non-zero slope for the negative part, which improves the results with a low-cost implementation [44].

In simplified terms considering only the convolution layer costs, we can formalize the baseline cost as

$$\mathcal{C}_{bs} = 4 * \mathcal{C}_v + \frac{\mathcal{C}_v}{D_k^2}. \tag{10}$$

B. EFFICIENT ARCHITECTURE

The efficient architecture combines KutralNext's architecture with inverted residual block [19], using the octave convolution [22] methodology. In this case, the octave convolution is separated into two regular convolutions, the *octave feature representation* or low-frequency convolution, which processes the most general features, and its counterpart or high-frequency convolution, which processes the most fine-grained features. A hyper-parameter α gives each convolution's size to process the signal further separately and exchange information at the end. We called this OctConvPN for the pointwise 1 × 1 convolution and OctConvDN for the depthwise convolution, where N denotes the number of output channels, as shown in Figure 3(a). The α parameter for both convolution blocks was settled to 0.5. The convolutional block is named KutralPlusBlockN-E (KPBN-E), where N-E refers to the number of output channels and the expansion rate t, respectively.

The efficient model, as presented in Figure 3(b), comprises an OctConvPN block, twins of batch-normalization, twins of LeakyReLU activation, OctConvDN block, twins of

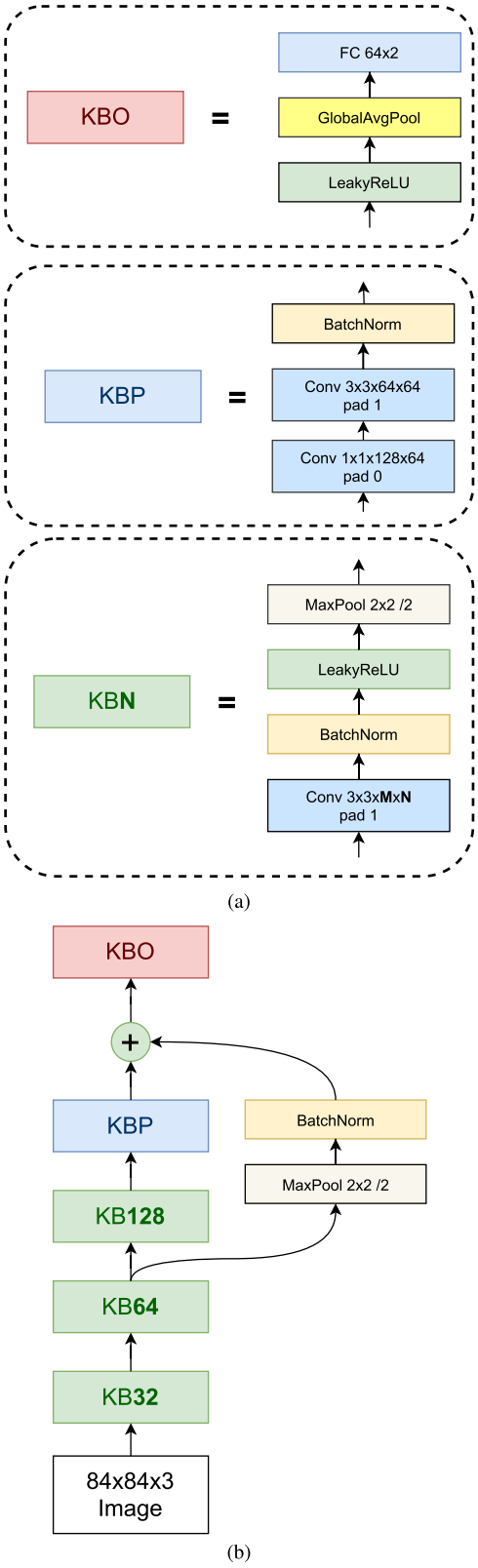


FIGURE 2. (a) The KutralNext main blocks are KutralBlockN (KBN), where *N* refers to the number of output channels, KutralBlockP (KBP), and the KutralBlockO (KBO). (b) the baseline KutralNext model with three KBN blocks, a KBP block, a shortcut connection, and a KBO block as output.

batch-normalization, twins of LeakyReLU activation, another OctConvPN, and finally, twins batch-normalization layers. For the first OctConvPN and the OctConvDN blocks, the N output channel number is given by multiplying the number of output channels and the expansion rate from the KPBN-E block. The architecture comprises a KBN block, followed by three KPBN-E blocks with a shortcut connection from the first KPB64-4 block. That process the signal through twins max-pooling layers, twins bath-normalization layers and, an OctConvPN block to the final KBO block.

In the case of the separable depthwise convolution used in the inverted residual block [19], the increasing number of parameters and the reduced flops number is still efficient, as demonstrated in equation (2) given the group way of processing the channels where $groups = C_{in}$ and $out\_channels = C_{in} * K$. In those groups, the output filters are K times the input filters, reducing the mathematical complexity of the operation as expressed in equation (4). In this regard, the model cost notation replaces the convolution layers cost expressed in (1), by the $\mathcal{C}_{irb}$ cost as follows

$$\mathcal{C}_{kpm} = \mathcal{C}_v + 3 * \mathcal{C}_{irb}, \tag{11}$$

obtaining a first reduction denoted by

$$\mathcal{W}_{kpm} = 3 * \left(\frac{2}{D_k^2} + \frac{1}{t * N} \right). \tag{12}$$

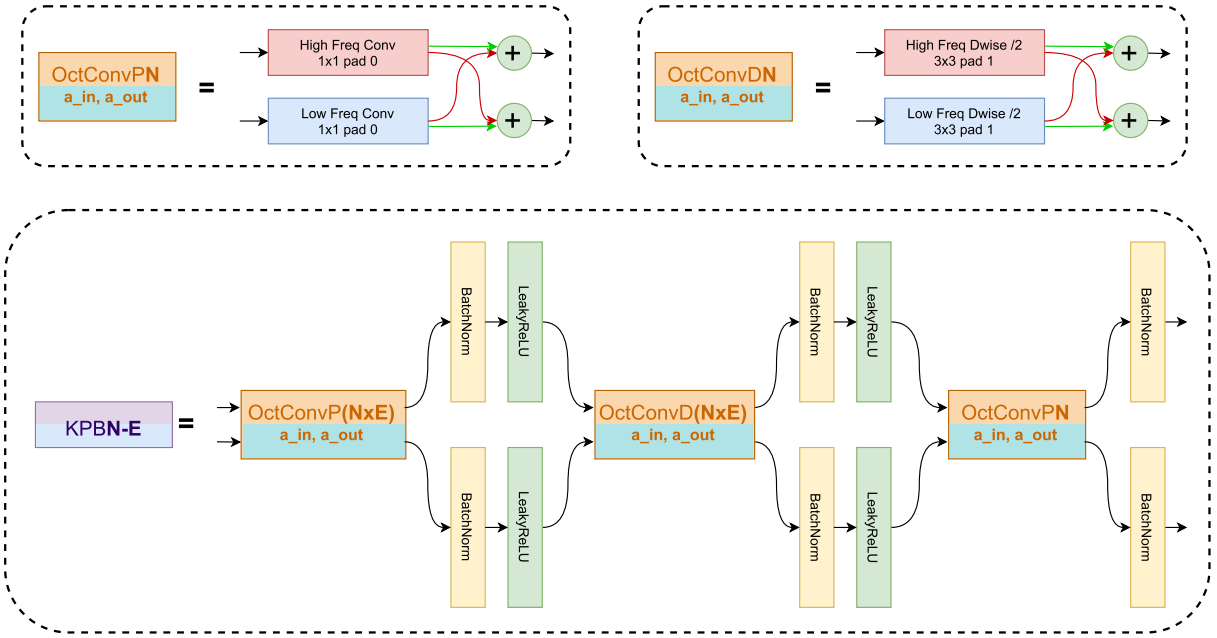
For the octave convolution, both parameters and flops are reduced. This strategy separates the filters processing on high and low frequency, computing the parameters information W into two components $W = [W_H, W_L]$ and exchanging information between them, expressed in equation (6). The KutralNext+ cost reduction, is given by (12) and (9)

$$\mathcal{W}_{kp} = \mathcal{W}_{kpm} * \frac{\mathcal{C}_o}{\mathcal{C}_v} + \frac{1}{D_k^2}. \tag{13}$$

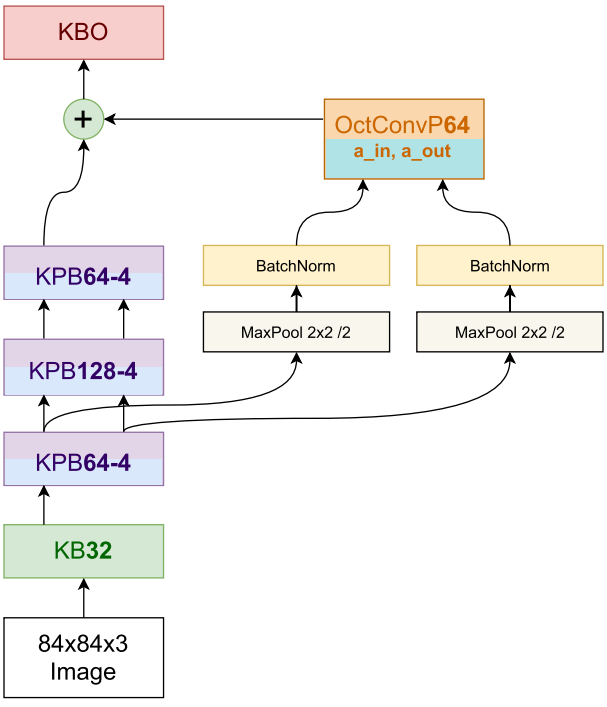
C. IMAGENET PRETRAINING

One of the challenges in deep learning model developments is the huge amount of data required for training. In this regard, using pretrained models over a challenging dataset with a considerable quantity of instances and labels improves the results using transfer learning and fine-tuning, reducing the data required to learn filter kernels and acquire valuable information from a high dimensional input.

For this purpose, we used the ImageNet ILSVRC 2012 dataset. It comprises 1.3 million instances with 1,000 classes, designed for a classification and detection competition, widely used as a model's performance benchmark. Many classical DL models, such as ResNet and EfficientNet, have been trained with ImageNet. They are publicly available in different repositories for the community. We used the ImageNet dataset for training the baseline, and the efficient architectures for later use in the fire and smoke classification task.



(a)



(b)

FIGURE 3. (a) The efficient model's main blocks, KutralPlusBlockN-E (KPBN-E), where *N* refers to the number of output channels and *E* for the *t* value of expansion rate. OctConvPN and OctConvDN are the octave convolution version block for the separable depthwise convolution. (b) The efficient model with a KBN block, three KPBN-E blocks, a shortcut connection, and a KBO block as output. The left-side connection from one block to another represents the $\alpha = 0$ value for the input or output of octave convolution.

D. SINGLE-LABEL AND MULTI-LABEL VARIANTS

All of the models were tested over two kinds of classification tasks. The first one was a fire-flame single-label classification task, where the models must indicate if a fire presence exists or not in the images. For this case, the fully-connected layer of the KBO block is composed of two outputs, one for the positive and the other for the negative cases of fire presence. The second one was a fire and smoke multi-label classification task, where the models must indicate if the image has fire, smoke, or nothing. In this case, the fully-connected

layer comprises two neurons. The first one indicates if a fire presence exists in the image, and the second indicates if smoke is present.

Each proposal presents two neurons in the fully-connected layer as output and uses the focal loss function and the sigmoid activation. Experiments have demonstrated that the multi-label approach, in addition to recognizing smoke in the image, also improves the model's capability to acquire fire features.

E. CLASS BALANCED LOSS

As a dataset grows, focused on obtaining more instances of those classes of interest, it is much more likely to have a long-tailed distribution with many underrepresented classes. A novel framework was implemented in our proposal to deal with this class imbalance issue. This framework uses the effective number of samples or expected volume of samples to define each class's impact on the loss value. This method is named class balanced loss [45], and it defines the effective number of samples as $(1 - \beta^n)/(1 - \beta)$, where n is the number of samples and β an hyper-parameter $\in [0, 1]$ which control how fast the effective number of samples grows as n increases. This loss function's main idea is to introduce a class weighting factor inversely proportional to the effective number of samples to balance the output loss value as a model- and loss-agnostic method, formulated as

$$\mathrm{CB}(p, y) = \frac{1 - \beta}{1 - \beta^{n_y}} \mathcal{L}(p, y), \tag{14}$$

where n_y is the number of samples for the class y, $\mathcal{L}(p, y)$ is the loss function for the predicted class probability p.

In our proposal, the $\mathcal{L}(p, y)$ loss function is replaced by the focal loss (FL) [42]. It is an α-weighted method to address the class imbalance issue, defining each class impact in the loss value with $\alpha \in [0, 1]$ for the target class y, and $1 - \alpha$ for the other classes, defined as follows

$$\mathrm{FL}(p_y) = -(1 - p_y)^{\gamma} \log(p_y), \tag{15}$$

where p_y is the probability of the y class, $(1 - p_y)^{\gamma}$ is a modulating factor with a $\gamma \geq 0$ hyper-parameter to determine how smoothly it affects the loss function, focusing in difficult samples. Each p_y class probability at the exit of the models is represented by the sigmoid cross-entropy loss denoted by

$$p_y = \frac{1}{(1 + \exp^{-z})}. \tag{16}$$

In this regard, our implementation included the base sigmoid cross-entropy loss (16), with the datasets classes weighted by the focal loss (15), and defining each class impact by the class balanced loss (14), formulated in next

$$\mathrm{CB}_{\mathrm{focal}}(z, y) = -\frac{1 - \beta}{1 - \beta^{n_y}} (1 - p_y)^{\gamma} \log(p_y), \tag{17}$$

where z is the model's predicted class probability.

IV. EXPERIMENTS

The environment used to train and test each model was an online open cloud platform for machine learning algorithms. This online platform provides a ready-to-use ecosystem with libraries for data manipulation, data visualization, and the training process, among others. The environment is available through a virtual machine configured with up to 13GB of memory, an Intel Xeon@2.30GHz, and an NVIDIA GPU with 12GB of memory.

A. DATASETS

Three publicly available datasets used were designed for a fire or smoke single-label classification task, with fire, smoke, or none classes, named FiSmo[1] [27], FireNet[2] [26], and FireSmoke.[3] All the datasets were previously used in fire and fire and smoke classification tasks as presented in [9], [26], [28], [38], [39]. For this project, 16,140 datasets' images were checked by one person, labeling all the images for a multi-label classification approach. Missing label addition was performed during review when both fire and smoke classes were present in the image mainly.

Four datasets emerged from the augmented and combined data once all the image labels were reviewed and corrected. From the three primary datasets, the FireNet and FiSmo datasets were used with the original 3, 296 and 6, 063 image instances, respectively. Additionally, FiSmo was augmented until a total of 6, 548 instances, and the FireSmoke dataset was used to complement FireNet. The FireNet dataset contains a test subset with 871 images, used for testing purposes for the fire-only recognition task. The augmentation of FiSmo used in this research adds 485 black images as none class because it has been shown to improve the model's performance for fire recognition [9], [38]. Finally, the combination of the FireNet and FireSmoke datasets were merged into a new one called KutralSmoke with 6, 296 images. This KutralSmoke dataset contains a test subset with 1,171 images used for testing the fire and smoke recognition task. This dataset was consolidated to get a training and testing subset with more instances labeled as smoke and reduce the class unbalancing. The instances allocation for training and testing of the datasets follows the implementation used in their original works, being FiSmo the dataset with training subset only.

For the single-label experiment, the FireNet (training), FiSmo, and FiSmoA datasets were used for training, summarizing 8, 973 images. The FireNet (testing) dataset was used for testing with 871 images. Only the fire and none classes were used from each dataset for this single-label experimentation. The fire and smoke labeled instances were considered with the fire class, and those smoke labeled instances were considered with the none class. For the

[1] https://github.com/mtcazzolato/dsw2017

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

[3] https://github.com/DeepQuestAI/Fire-Smoke-Dataset

TABLE 1. Quantity of images per class present in each dataset.

Dataset	Set	Fire & Smoke	Fire	Smoke	None	Total
FireNet	training	750	352	46	1,277	2,425
FireNet	testing	55	537	1	278	871
FireSmoke	training	677	247	862	914	2,700
FireSmoke	testing	64	39	93	104	300
KutralSmoke	training	1,427	599	908	2,191	5,125
KutralSmoke	testing	119	576	94	382	1,171
FiSmo	training	795	1,267	384	3,617	6,063
FiSmoA	training	795	1,267	384	4,102	6,548
Total		3,146	3,331	1,433	8,230	16,140

multi-label experiment, 11, 188 images were used for training, considering the FiSmo and KutralSmoke (training) datasets, and 1, 171 images were used for testing from the KutralSmoke (testing) dataset. For this last experiment, all labels were used with no changes. More details about the datasets class' distribution are shown in Table 1, and image samples are in Figure 4. During each experiment's parameters' optimization, the training datasets were split for training and validation subsets, considering the 20% as validation and the other 80% for training.

B. MODELS

All of the models were trained and tested with all of the corresponding datasets mentioned earlier in Section IV-A. For our models previously trained over Imagenet, the training process refers to transfer-learning and fine-tuning the models' weights for this new data distribution for recognizing fire and smoke. The training and testing processes were executed five times to get statistical significance. Considering the five models and that training is highly time-consuming, we hypothesized that five executions would show a tendency and variability in the metrics obtained for each model in the different tasks. Each training execution was iterated over 100 epochs with a batch size of 32 instances each. For our case, just KutralNext presents a variation in the learning rate starting at 10^{-3} and reduced to 10^{-4} after epoch 85 to avoid overfitting the parameters and stabilizing the learning in the last epochs. All the other models use a fixed learning rate of 10^{-3}.

C. PERFORMANCE EVALUATION

The experimental setup compared the fire and smoke recognition performance of our KutralNext architectures against fire-specialized models. The first one compares the single-label fire classification task performance, where each model must infer the presence or not of fire in images. For this case, just the fire label is used from the image datasets, codifying the target label y into a two-component vector $\in [0, 1]$. When the first component is equal to 1, it indicates no fire presence, and when the second component is 1, it indicates fire presence. The sum of the outputs, in this case, must be equal to 1. The second experiment is the multi-label fire and smoke classification task, where each model must infer the presence of fire, smoke, or none in images. For this task, the fire and smoke labels were used from each dataset, codifying the target label y as vector $\in [0, 1]$, with one component for each class. In this case, the target label can represent the fire and smoke presence, with both components equal 1, and both components equal 0, representing neither fire nor smoke presence. The behavior was checked under two different data representations and distribution in both experiments, using a cross-dataset test, proving each model's robustness.

TABLE 2. The computational cost of each implemented model represented as flops and parameters.

Model	Input Size	Flops	Parameters
KutralNext	84x84	76.85M	138.91K
KutralNext+	84x84	24.59M	185.25K
FireDetection [39]	224x224	783.50M	335.53K
FireNet [26]	64x64	8.94M	646.82K
OctFiResNet [38]	96x96	928.95M	956.23K

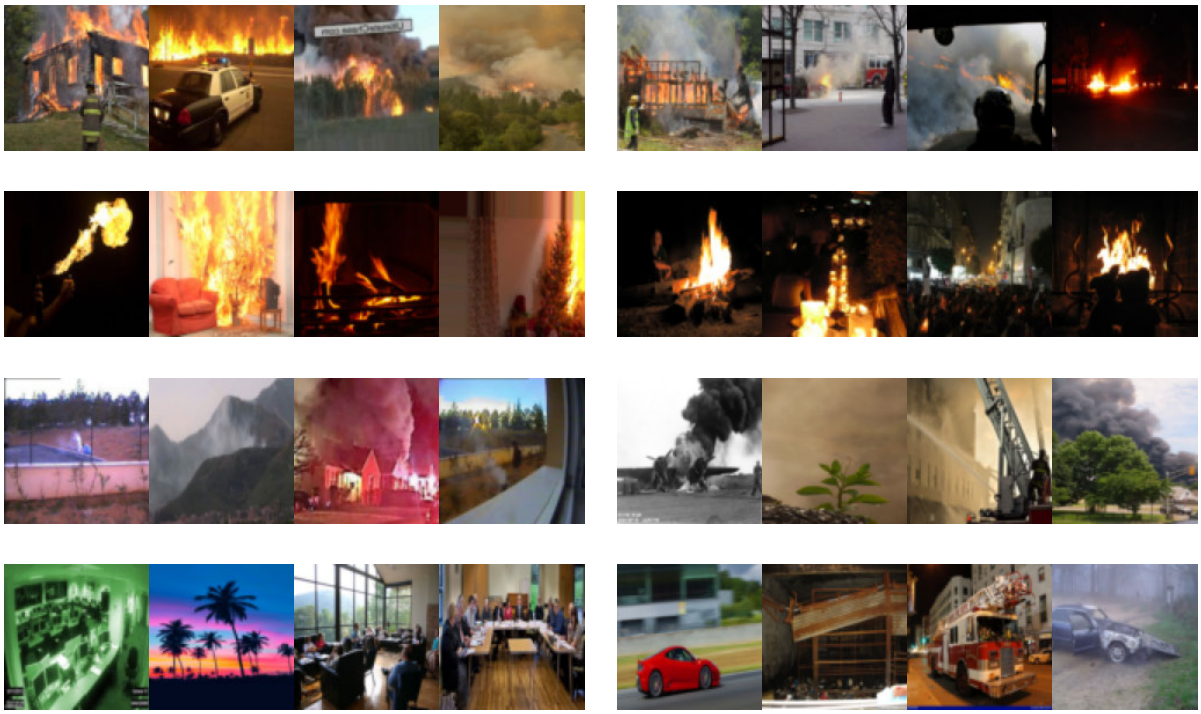
All of the images were preprocessed with a resize transformation to fit each model's input size and normalized with values $\in [0, 1]$. For each model, a different loss function was used to follow their original implementation. For FireNet, OctFiResNet, and FireDetection, a cross-entropy loss was used with a softmax activation in the single-label experimentation and a binary cross-entropy loss with sigmoid activation in the multi-label approach.

V. RESULTS

All of the models were trained and tested against FiSmo, FireNet, and KutralSmoke datasets to compare their suitability in single- and multi-label recognition tasks. More than one data source under different distribution is affordable to check the model's generalization capability to acquire features and recognize fire or smoke presence in images. Thus, as aforementioned, two types of experiments were carried out.

The first subsection explains the results obtained during the single-label fire recognition task. The second subsection describes each model's recognition of fire and smoke in still images as a multi-label classification approach. The third section discusses how well the models could generalize in both tasks and the benefits presented by our KutralNext+ model. All of the models were compared using the following metrics: the accuracy obtained during validation and testing, the receiver operating characteristic (ROC) curve, the area under the ROC curve (AUROC), the number of floating-point operations (flops), precision, recall, f1-score, and the time required to process all the images in the corresponding testing dataset.

Table 2 presents the costs of each model in terms of parameters and flops. Tables 3 and 5 display the average training results obtained during the five executions with their standard deviation values for the single- and multi-label approaches, respectively.



(a)

(b)

(c)

FIGURE 4. Data samples for each dataset used in this work. (a) FireNet image samples, (b) FiSmo image samples, (c) KutralSmoke image samples. In the first row are the Fire and Smoke labeled images, the second row are the fire labeled images, the third row are the smoke labeled image, and in the bottom row are the none labeled images.

A. SINGLE-LABEL CLASSIFICATION: FIRE RECOGNITION

The first experiment considers our proposals' single-label fire recognition task performance compared with state-of-the-art fire recognition deep learning models. Table 3, presents the training and testing statistics results. Table 4 demonstrates each model's performance statistics over all the datasets

with averaged values and standard deviation. Overall, our KutralNext+ can generalize better for this task, and KutralNext achieves better performance acquiring fire features, both against previous fire recognition models. Our proposals achieved the best results. They were the most time inexpensive models in image processing.

From the results presented in Table 3, previous fire recognition models' results are similar to OctFiResNet as the best accurate model during validation, followed by FireDetection and FireNet. Nevertheless, in test accuracy, FireDetection performs better than OctFiResNet. Our proposals have proven to achieve the best generalization trained over different data distributions as FiSmo and FiSmoA, where KutralNext+ obtained the best mean validation and test accuracy. In terms of time, OctFiResNet is the model that requires more time to deal with the test data taking about 3 seconds to process the images, followed by FireDetection with 1.5 seconds, with KutralNext and KutralNext+ requiring 0.45 and 0.42 seconds, respectively. The FireNet model presents a lightweight approach found in the literature and performs 29% faster in processing the test dataset than KutralNext+ with 0.30 seconds. However, it presents a difference of 4.96% and 14.15% in validation and test accuracy, respectively. All of the KutralNext models outperform all the previous fire recognition models in a single-label approach, with KutralNext+ in the first position, followed by KutralNext, demonstrating an efficient computational cost to recognize fire. The best trained average KutralNext+ model is 0.29% and 5.51% higher than KutralNext. It is 1.83% and 2.65% higher than OctFiResNet in terms of validation and test accuracy, respectively.

Moreover, in Table 4, in terms of fire's feature acquisition for each model, the FireNet model obtains the best AUROC and precision among previous fire recognition models, followed by OctFiResNet and FireDetection. In this regard, it is demonstrated that a few layers can acquire enough features to recognize fire. Nevertheless, it is ineffective to process more complex images where the fire is present. KutralNext performs the best in both AUROC and precision values for all datasets, achieving 94.00% and 97.13%, respectively. KutralNext+ performs competitive results against KutralNext, with 93.64% and 97.03% for AUROC and precision, respectively, all mean values. In this regard, a demonstration between the trade-off and the model's depth is achieved, with efficient results recognizing fire. Additionally, the KutralNext+ presents a similar result under an efficient configuration processing in less time an image.

Interesting results were obtained for the black augmented FiSmo version, FiSmoA, which improved FireNet, KutralNext, KutralNext+, and remarkably OctFiResNet in test accuracy values. For FireDetection, the augmentation negatively impacted the performance, reducing by 2% test accuracy, and it increased the deviation value compared to FiSmo. This result was presented for test performance results in Table 4, where the FiSmoA dataset with black images increases the performance for all the models, resulting in a reduction over the standard deviation values.

Figure 5(a) shows more detailed performance for the models trained over FireNet where KutralNext+ obtained 97.59%, followed by KutralNext with 94.18% AUROC index. In the ROC curve, KutralNext performs well at a low false-positive rate. The use of the FireNet dataset proves the model's generalization over the same data distribution. A different data distribution model's behavior is achieved with FiSmo and FiSmoA showed in Figures 5(b) and 5(c). In FiSmo, KutralNext retains first place with 92.44% of AUROC index, followed by KutralNext+ with 90.57%.

An even higher AUROC value was obtained for the augmented version of FiSmo, where KutralNext achieves first place with 95.39%, followed by our KutralNext+ model with 92.79%. KutralNext achieved the best ROC curve with low false-positive rates in both cases.

B. MULTI-LABEL CLASSIFICATION: FIRE AND SMOKE RECOGNITION

In this second experiment, the performance in the fire and smoke multi-label recognition task for our models' proposals was checked. Two datasets were used for training and one dataset for testing. The training datasets were FiSmo and KutralSmoke. The testing dataset was the KutralSmoke Test subset. The models' performance was compared, optimizing its parameters with different data complexity and distribution of the corresponding labels to check its generalization capability. Due to the chance of fire and smoke appearing in the same image, the fire and smoke classification task was addressed under a multi-label setup. Table 5 shows the statistical results for each model trained over all the datasets with averaged values for the validation, test accuracy, and test time. Table 6 presents the test performance for each model. Our proposals are the best for recognizing fire and smoke as the most accurate and time inexpensive models. The classification was considered binary, with fire, smoke, or both classes as a true label, and none class as a false label.

The models' training performance in terms of accuracy are shown in Table 5. For the mean validation accuracy, KutralNext performs the best with an 85.43%, followed by KutralNext+ with an 85.84%, considering their deviation values. Different results were obtained during testing. KutralNext+ performs the best under the same and different data distribution with an 81.53%. Our models surpass the state-of-the-art fire recognition models, requiring less time in processing the test data images.

In terms of time required to process the 1,171 testing images, OctFiResNet is the most time-consuming, taking 2.0 seconds, followed by FireDetection with 1.87 seconds. For the KutralNext architectures, KutralNext+ is the model that requires more time with 0.61 seconds, leaving KutralNext as the model that requires less time with 0.41 seconds. FireNet is a model that requires less time to process the images. Nevertheless, it also presents the lowest mean validation and test accuracy.

TABLE 3. Training results during 5 executions in the fire recognition task.

DS	Model	Val. acc.	Test acc.	Test (ms)
FireNet	FireDetection [39]	94.99% ± 0.69%	86.59% ± 2.62%	1550 ± 26
	FireNet [26]	93.59% ± 0.45%	82.62% ± 4.43%	**301 ± 17**
	KutralNext	98.36% ± 0.38%	79.06% ± 4.61%	458 ± 27
	KutralNext+	**98.40% ± 0.22%**	**89.53% ± 2.66%**	414 ± 20
	OctFiResNet [38]	97.04% ± 0.61%	82.73% ± 6.12%	3191 ± 39
FiSmo	FireDetection [39]	90.77% ± 0.21%	73.04% ± 3.67%	1543 ± 24
	FireNet [26]	87.63% ± 0.32%	62.69% ± 7.98%	**302 ± 20**
	KutralNext	92.03% ± 0.17%	77.43% ± 2.60%	450 ± 20
	KutralNext+	**92.44% ± 0.37%**	**80.62% ± 1.26%**	428 ± 14
	OctFiResNet [38]	90.21% ± 0.49%	70.70% ± 5.05%	3156 ± 46
FiSmoA	FireDetection [39]	88.49% ± 0.20%	71.23% ± 5.22%	1531 ± 12
	FireNet [26]	85.44% ± 0.61%	64.20% ± 2.76%	**287 ± 21**
	KutralNext	90.72% ± 0.09%	78.05% ± 3.38%	443 ± 12
	KutralNext+	**90.72% ± 0.09%**	**82.92% ± 2.57%**	409 ± 11
	OctFiResNet [38]	88.53% ± 0.47%	76.37% ± 8.02%	3177 ± 34
Average	FireDetection [39]	91.42% ± 0.37%	76.95% ± 3.84%	1541 ± 21
	FireNet [26]	88.89% ± 0.46%	69.84% ± 5.06%	**297 ± 19**
	KutralNext	93.70% ± 0.21%	78.18% ± 3.53%	450 ± 20
	KutralNext+	**93.85% ± 0.23%**	**84.36% ± 2.17%**	417 ± 15
	OctFiResNet [38]	91.93% ± 0.52%	76.60% ± 6.40%	3175 ± 39

TABLE 4. Test performance during 5 executions in the fire recognition task.

DS	Model	AUROC	Precision	Recall	F1-score
FireNet	FireDetection [39]	91.25% ± 2.50%	93.79% ± 1.06%	86.01% ± 4.91%	89.65% ± 2.30%
	FireNet [26]	92.90% ± 4.83%	94.78% ± 3.66%	78.99% ± 7.54%	85.92% ± 4.19%
	KutralNext	94.18% ± 2.61%	95.60% ± 0.77%	73.21% ± 6.35%	82.81% ± 4.25%
	KutralNext+	**97.59% ± 1.05%**	**96.23% ± 1.02%**	**89.09% ± 4.12%**	**92.47% ± 2.11%**
	OctFiResNet [38]	91.07% ± 5.59%	91.85% ± 3.32%	81.76% ± 6.70%	86.46% ± 5.08%
FiSmo	FireDetection [39]	81.16% ± 3.81%	92.55% ± 2.87%	65.61% ± 4.14%	76.75% ± 3.54%
	FireNet [26]	86.44% ± 3.35%	95.48% ± 4.10%	47.57% ± 12.55%	62.59% ± 11.49%
	KutralNext	**92.44% ± 1.49%**	**97.22% ± 1.11%**	69.53% ± 4.78%	80.98% ± 2.92%
	KutralNext+	90.57% ± 2.00%	97.25% ± 1.51%	**74.09% ± 2.46%**	**84.07% ± 1.37%**
	OctFiResNet [38]	80.03% ± 5.01%	94.71% ± 1.87%	60.21% ± 7.03%	73.45% ± 5.65%
FiSmoA	FireDetection [39]	83.02% ± 6.79%	94.24% ± 3.68%	61.42% ± 6.78%	74.22% ± 5.55%
	FireNet [26]	92.35% ± 1.37%	97.79% ± 0.82%	48.45% ± 4.37%	64.69% ± 3.77%
	KutralNext	**95.39% ± 1.26%**	**98.57% ± 0.79%**	69.02% ± 4.93%	81.10% ± 3.39%
	KutralNext+	92.79% ± 2.22%	97.62% ± 1.68%	**77.16% ± 3.34%**	**86.16% ± 2.24%**
	OctFiResNet [38]	92.09% ± 3.22%	97.24% ± 1.50%	67.10% ± 11.63%	78.98% ± 8.25%
Average	FireDetection [39]	85.15% ± 4.37%	93.53% ± 2.54%	71.01% ± 12.19%	80.21% ± 7.93%
	FireNet [26]	90.56% ± 3.18%	96.02% ± 2.86%	58.23% ± 17.19%	71.07% ± 12.88%
	KutralNext	**94.00% ± 1.79%**	**97.13% ± 0.89%**	70.59% ± 5.36%	81.63% ± 3.41%
	KutralNext+	93.65% ± 1.76%	97.03% ± 1.40%	**80.11% ± 7.39%**	**87.57% ± 4.11%**
	OctFiResNet [38]	87.73% ± 4.61%	94.60% ± 2.23%	69.69% ± 12.33%	79.64% ± 8.15%

A general overview of each model's metrics over the test dataset are shown in Table 6. In the first place, for the fire label, KutralNext demonstrated the best average AUROC value and OctFiResNet the best mean precision value in this multi-label test approach. Considering the mean AUROC between both datasets, the KutralNext model obtains a 94.47% index value, taking first place, followed by KutralNext+ with 93.40% index value. Overall, all of the models present a good performance in detecting fire under this approach. For the smoke label, a lower outcome is shown in AUROC and precision terms, in the second place. KutralNext+ model achieves a remarkable AUROC of 89.59% and precision of 56.27%. KutralNext attained second place with 87.00% and 46.92% for the same metrics. Our proposals are the best models in acquiring smoke features under a multi-label approach compared to the previous one. Overall, the models have shown a better outcome trained over the same data distribution than a different one.

Figure 6 shows the mean ROC values obtained for the models trained over all of the datasets to compare each model's features acquisition performance. Our proposals presented

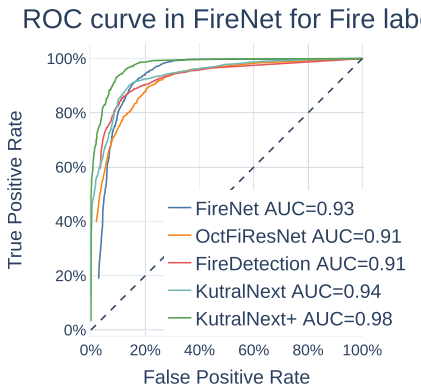
TABLE 5. Models' training results during 5 executions in the fire and smoke recognition task.

DS	Model	Val. acc.	Test acc.	Test (ms)
KutralSmoke	FireDetection [39]	80.95% ± 0.92%	77.59% ± 3.22%	1883 ± 81
	FireNet [26]	78.85% ± 0.46%	77.11% ± 3.60%	**339 ± 23**
	KutralNext	89.66% ± 0.44%	86.70% ± 2.02%	430 ± 21
	KutralNext+	**90.46% ± 0.48%**	**88.08% ± 0.69%**	603 ± 34
	OctFiResNet [38]	84.00% ± 0.63%	79.03% ± 4.58%	2040 ± 11
FiSmo	FireDetection [39]	77.18% ± 0.99%	63.04% ± 8.60%	1856 ± 99
	FireNet [26]	74.70% ± 0.66%	56.89% ± 6.26%	**335 ± 22**
	KutralNext	**81.20% ± 0.74%**	71.36% ± 2.31%	424 ± 21
	KutralNext+	81.22% ± 3.18%	**74.98% ± 3.22%**	624 ± 33
	OctFiResNet [38]	78.04% ± 0.41%	56.69% ± 2.38%	2046 ± 6
Average	FireDetection [39]	79.07% ± 0.96%	70.32% ± 5.91%	1870 ± 90
	FireNet [26]	76.77% ± 0.56%	67.00% ± 4.93%	**337 ± 23**
	KutralNext	**85.43% ± 0.59%**	79.03% ± 2.16%	427 ± 21
	KutralNext+	85.84% ± 1.83%	**81.53% ± 1.96%**	614 ± 33
	OctFiResNet [38]	81.02% ± 0.52%	67.86% ± 3.48%	2043 ± 9

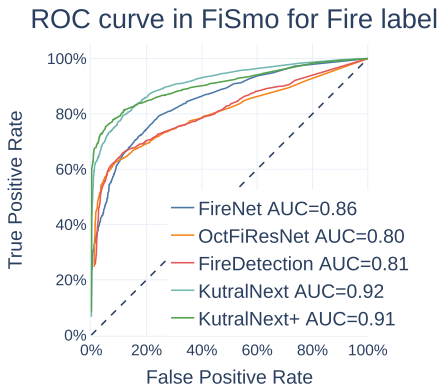
TABLE 6. Performance during 5 executions in fire and smoke recognition task.

DS	Model	AUROC	Precision	Recall	F1-score
			Fire Label		
KutralSmoke	FireDetection [39]	88.72% ± 4.04%	95.02% ± 1.87%	71.51% ± 9.85%	81.23% ± 6.03%
	FireNet [26]	94.18% ± 1.68%	94.07% ± 1.28%	75.94% ± 6.40%	83.95% ± 4.33%
	KutralNext	96.96% ± 0.49%	**97.12% ± 0.80%**	80.14% ± 2.44%	87.80% ± 1.34%
	KutralNext+	**97.46% ± 0.43%**	96.69% ± 1.21%	**84.17% ± 2.70%**	**89.99% ± 1.85%**
	OctFiResNet [38]	94.84% ± 2.67%	94.74% ± 2.11%	79.22% ± 8.63%	86.05% ± 5.37%
FiSmo	FireDetection [39]	85.73% ± 7.74%	92.61% ± 4.77%	55.60% ± 11.89%	69.00% ± 10.49%
	FireNet [26]	83.66% ± 5.49%	90.74% ± 3.48%	43.20% ± 7.95%	58.26% ± 7.81%
	KutralNext	**91.98% ± 2.97%**	93.64% ± 4.11%	**75.74% ± 2.70%**	**83.70% ± 2.41%**
	KutralNext+	89.35% ± 2.03%	91.74% ± 3.48%	70.68% ± 5.79%	79.64% ± 3.13%
	OctFiResNet [38]	84.25% ± 3.61%	**96.70% ± 2.03%**	54.01% ± 5.71%	69.12% ± 4.08%
Average	FireDetection [39]	87.23% ± 5.89%	93.82% ± 3.32%	63.55% ± 13.28%	75.11% ± 10.33%
	FireNet [26]	88.92% ± 3.59%	92.40% ± 2.38%	59.57% ± 18.55%	71.11% ± 14.79%
	KutralNext	**94.47% ± 1.73%**	95.38% ± 2.45%	**77.94% ± 3.36%**	**85.75% ± 2.84%**
	KutralNext+	93.40% ± 1.23%	94.22% ± 2.35%	77.43% ± 8.29%	84.81% ± 5.97%
	OctFiResNet [38]	89.54% ± 3.14%	**95.72% ± 2.07%**	66.62% ± 14.97%	77.58% ± 9.99%
			Smoke Label		
KutralSmoke	FireDetection [39]	70.78% ± 3.84%	29.30% ± 2.59%	78.22% ± 4.91%	42.56% ± 2.89%
	FireNet [26]	72.22% ± 1.55%	28.00% ± 1.73%	76.43% ± 2.40%	40.95% ± 1.76%
	KutralNext	91.74% ± 1.23%	**52.91% ± 3.82%**	84.41% ± 3.10%	**64.94% ± 2.74%**
	KutralNext+	**92.59% ± 1.77%**	52.19% ± 6.55%	**86.01% ± 2.26%**	64.70% ± 4.82%
	OctFiResNet [38]	76.42% ± 6.25%	31.49% ± 5.90%	83.29% ± 2.12%	45.45% ± 5.77%
FiSmo	FireDetection [39]	67.38% ± 3.92%	33.06% ± 3.21%	41.41% ± 6.19%	36.57% ± 3.52%
	FireNet [26]	67.79% ± 5.35%	35.01% ± 6.30%	41.50% ± 5.38%	37.74% ± 5.17%
	KutralNext	82.27% ± 1.19%	40.93% ± 2.41%	**65.63% ± 2.45%**	50.38% ± 2.18%
	KutralNext+	**86.59% ± 3.22%**	**60.35% ± 12.66%**	56.62% ± 6.48%	**57.30% ± 5.92%**
	OctFiResNet [38]	66.95% ± 4.95%	29.75% ± 3.68%	42.72% ± 14.47%	34.36% ± 6.92%
Average	FireDetection [39]	69.08% ± 3.88%	31.18% ± 2.90%	59.81% ± 20.10%	39.56% ± 4.38%
	FireNet [26]	70.00% ± 3.45%	31.51% ± 4.02%	58.97% ± 18.82%	39.34% ± 4.01%
	KutralNext	87.00% ± 1.21%	46.92% ± 3.11%	**75.02% ± 10.24%**	57.66% ± 8.02%
	KutralNext+	**89.59% ± 2.50%**	**56.27% ± 9.60%**	71.32% ± 16.15%	**61.00% ± 6.41%**
	OctFiResNet [38]	71.69% ± 5.60%	30.62% ± 4.79%	63.01% ± 23.50%	39.90% ± 8.38%

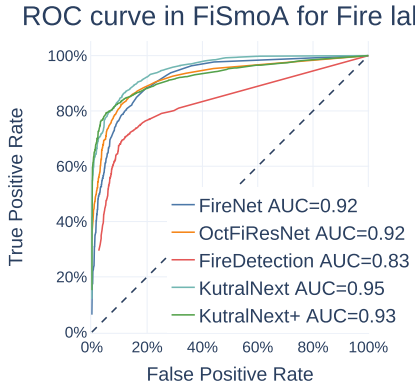
the best results for both classes from the datasets we used, capable of acquiring features at a low false-positive rate. Important results were obtained for the smoke label compared with previous models, as shown in Figures 6(c) and 6(d). Additionally, KutralNext and KutralNext+ obtained the best results under a different data distribution as the case for the FiSmo dataset. Thus, our proposals demonstrated their implemented techniques efficiency because the models' designs were not meant to recognize smoke. Even so, it achieved the best results in recognizing smoke.



(a) Average ROC curve obtained for the models trained over FireNet. Can be observed that KutralNext+ achieves the highest AUC value. Additionally, KutralNet performs the second-best surpassing the FireNet model in the third pace.



(b) Average ROC curve obtained for the models trained over FiSmo. Can be observed that KutralNext achieves the highest AUC value, followed by KutralNext+ and FireNet.



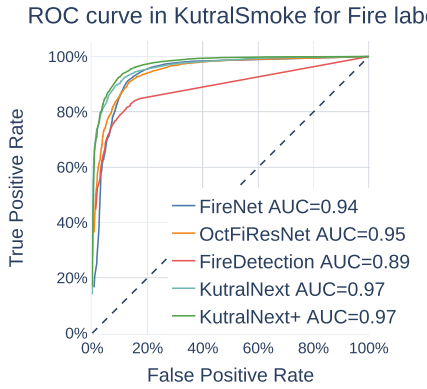
(c) Average ROC curve obtained for the models trained over FiSmoA. It is shown that KutralNet achieves the best AUC score, followed by KutralNext+ and FireNet.

FIGURE 5. Single-label classification average ROC curves of five executions for the models trained over (a) FireNet, (b) FiSmo, and (c) FiSmoA datasets. As illustrated, our KutralNext+ outperforms the other models with the same data distribution in the FireNet dataset (a). It is competitive with KutralNext in different data distributions as (a) FiSmo and (b) FiSmoA. The obtained results between FireNet against OctiFiResNet and FireDetection prove that a few layers are required to acquire fire features.

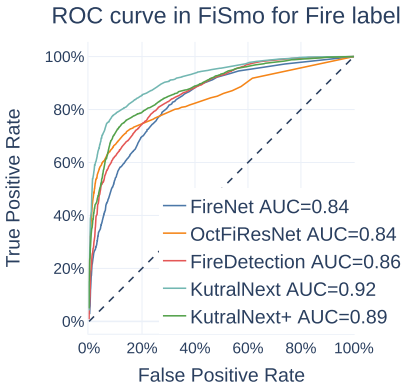
C. DISCUSSION

Our KutralNext deep learning model proposals were capable of achieving a proper performance for fire and smoke recognition as a single- and multi-label approach, compared to previous deep learning models in the same approach. For this research project, all compared previous models were designed under a single-label approach and adapted to be used under a multi-label approach. The FireDetection model was the only one designed to recognize fire and smoke from images. All of the models previously used were designed to recognize fire only. However, the output layer was successfully adapted for those models, demonstrated in the results obtained for the fire label. The central aspect of addressing a multi-label approach in still images is that fire or smoke can be present separately, together, or not be in the image. In this regard, the multi-label approach could be suitable for an early alert system to measure the fire's magnitude. This magnitude could be translated from the inference of each label present in the image. When only the smoke label was detected, it could be an early fire stage. When just the fire label was detected, it could be a fire of minor intensity. Alternatively, a more extensive fire was detected if both fire and smoke labels were present.

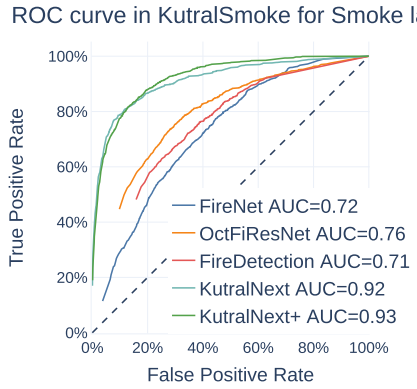
Moreover, the results highlight the importance of using a pre-trained model with ILSVRC2012 over one trained from scratch. This benefit was demonstrated by the KutralNext+ model's performance, from which the pre-trained versions perform significantly better than the one from-scratch version in the single-label fire classification task with a 5.10% more in average test accuracy. Additionally, present 4.05% less mean standard deviation. During the efficient models' training over the ILSVRC2012 dataset, the validation accuracy was not appropriate to classify the 1000 contained classes. Nevertheless, the optimized parameters obtained



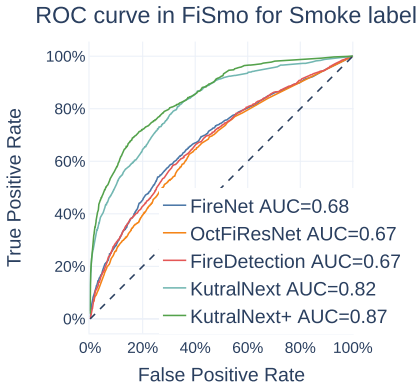
(a) Average ROC curve for fire label obtained from the models trained over KutralSmoke. Both KutralNext approaches present the same AUC score, however, KutralNext+ performs a higher number of images correctly classified. The OctFiResNet and FireNet are the following models.



(b) Average ROC curve for fire label obtained from the models trained over FiSmo. KutralNext outcome the highest AUC score, followed by KutralNext+ and FireDetection.



(c) Average ROC curve for smoke label obtained from the models trained over KutralSmoke. KutralNext+ achieves the highest AUC score, being the KutralNext proposals the only ones that outperform over 90% — followed by OctFiResNet and FireNet.



(d) Average ROC curve for smoke label obtained from the models trained over FiSmo. KutralNext+ obtained the best score, followed by KutralNext and FireNet.

FIGURE 6. Multi-label ROC curve average performance of five executions for the models trained over (a) KutralSmoke, and (b) FiSmo datasets. The KutralNext architecture presents the highest AUC value in both labels from both datasets. Exciting results are shown under a different data distribution, (b) and (d), where KutralNext performs well for the fire label and KutralNext+ for the smoke label. Additionally, it can be observed that previous approaches still lack generalization for the fire label and present a low performance processing the smoke label.

during the training of our efficient models, KutralNext and KutralNext+, were enough to obtain better performance for both fire and smoke recognition task-specific model architecture design. These from-scratch results were not included in this research. However, this aspect has been widely demonstrated [46]. Additionally, the portable version with the inverted residual block and the octave convolution methods reduced the model's flops. It improved accuracy in single- and multi-label fire and smoke recognition tasks, suitable for a portable device at a high frame rate. Thus, our portable proposal is suitable for a fire detection vision-based system for fire or smoke presence incidents.

Let us compare the models' computational cost in flops and allocation size as the number of parameters. This research demonstrated that the kind of convolution and its configuration define the model's size and complexity. Some existing convolution methods are more efficient than others in processing the input signal, requiring minimal computation resources to achieve remarkable performance. In this regard, FireNet uses fewer flops given its image resolution than KutralNext but presents more parameters. FireDetection requires the biggest image resolution as input but uses fewer parameters than FireNet and OctFiResNet, and fewer flops than OctFiResNet, demonstrating the use of other convolution configurations such as squeeze and expand from SqueezeNet [41]. Between our proposals, KutralNext+ presents fewer flops than KutralNext, but with more parameters. This effect is occasionated by the inverted residual block method, which adds more parameters given the separable depthwise convolution, but with reduced complexity

given the pointwise convolution. In this way, an efficient fire and smoke recognition model with enough parameters to generalize fire and smoke features was developed. This low-complexity configuration was only achieved with this specific-purpose architecture because general-purpose models require a huge amount of parameters to learn how to process features for a significant number of classes.

Hereof, our proposals are suitable to assess a fire accident, allowing the possibility to automate a response, controlling its propagation. Additionally, it can be used in a surveillance monitoring system with multiple cameras requiring low-cost computation hardware to recognize a fire event and location. The only concern encountered in this research, is the unrelated values of flops and the time required by each model. This issue is not related to the model or its techniques, but it is related to the PyTorch library.[4] Another considerable aspect of time is in the multi-label problem with the label preparation, which requires a few steps of encoding the classes before being processed by the model. Thus, the time issue can be solved for a final portable detection system migrating the library and implementing a specific label codification system.

Recently published fire and smoke classification works present different approaches to address the problem. One of them uses feature descriptors presented by Sari *et al.* [47], where the authors use the histogram of oriented gradient (HOG) to further classify fire with a support vector machine (SVM). Singh *et al.* [48] presented a CNN to classify video frame sequences into the fire, smoke, and fire and smoke classes as a single-label classification task. Their work achieves a high recognition rate related to the video dataset used and the similarity between frames, as demonstrated by Ayala *et al.* [38]. Some application works are presented by Altowaijri *et al.* [49], where an IoT system captures an image and sends it to be processed by a CNN in the cloud. Rahmatov *et al.* [50] use a visual fire detection system to route planning in a UAV system as industrial surveillance from fire hazards. In this regard, current works are still improving the accuracy or applications without improving the model's efficiency.

VI. CONCLUSION

Different kinds of industries present specific risks to experience a fire incident. The ones that manipulate fuel elements are the most exposed. In such cases, using preventive and reactive control methods reduces fire accidents. Hereof, efficient methods to monitor the facilities and rapidly control this kind of event by detecting fire or smoke in multiple places at low cost become essential.

A novel KutralNext approach for fire and smoke recognition was proposed with 138.9K parameters and 76.9M flops, with an efficient model developed in this research. KutralNext+ considerably reduces the number of flops to 24.6M, achieving the best performance with 84.36%, and 81.53% mean test accuracy in the fire and fire and smoke recognition tasks, respectively. Additionally, it comprises 97% fewer flops and 16% more accurate during fire and smoke recognition testing than FireDetection. Hence, it is executed 4x faster with better generalization.

Addressing the fire and smoke recognition model in a multi-label approach is affordable to implement a more specific early vision-based alert system. Nevertheless, our efficient proposals could recognize the smoke in images, even when the architecture design was not intended for this task. Additionally, the pre-training over the ILSVRC dataset, the convolution techniques, and the multi-label approaches considerably improve our specific-purpose models' performance. Therefore, our KutralNext+ achieved the best test metrics, generalizing fire and smoke labels more effectively, suitable for portable device implementations.

For future studies, we recommend improving the smoke label recognition under an efficient approach. Maybe the architecture could be adapted into an ensemble model, reducing the number of layers and sharing important features to infer. Furthermore, we plan to extend our research to fire and smoke detection using a bounding box approach. Additionally, we consider KutralNext and KutralNext+'s implementation in an embedded platform and check its real-time performance. Positive results also could lead to implementing KutralNext architecture in a reinforcement learning framework to control an UAV searching for fire, as presented in [50].

[4]Some users reported the slow implementation in the depthwise convolution using CUDA 32 bits floating-point operations to the PyTorch repository https://github.com/pytorch/pytorch/issues/18631

REFERENCES

[1] X. Úbeda and P. Sarricolea, "Wildfires in chile: A review," *Global Planet. Change*, vol. 146, pp. 152–161, Nov. 2016.
[2] M. O. Nawaz and D. K. Henze, "Premature deaths in Brazil associated with long-term exposure to $PM_{2.5}$ from Amazon fires between 2016 and 2019," *GeoHealth*, vol. 4, no. 8, Aug. 2020, Art. no. e2020GH000268.
[3] J. Moreno, M. Arianoutsou, A. González-Cabán, F. Mouillot, W. Oechel, D. Spano, K. Thonicke, V. Vallejo, and R. Vélez, "Forest fires under climate, social and economic changes in Europe, the Mediterranean and other fire-affected areas of the world," *FUME Lessons Learned Outlook*, vol. 1, no. 1, pp. 7–8, Jan. 2014.
[4] A. M. Gill, "Landscape fires as social disasters: An overview of 'the bushfire problem,'" *Environ. Hazards*, vol. 6, no. 2, pp. 65–80, Jan. 2005.
[5] L. Ding, F. Khan, and J. Ji, "Risk-based safety measure allocation to prevent and mitigate storage fire hazards," *Process Saf. Environ. Protection*, vol. 135, pp. 282–293, Mar. 2020.
[6] B. Kong, Z. Li, E. Wang, W. Lu, L. Chen, and G. Qi, "An experimental study for characterization the process of coal oxidation and spontaneous combustion by electromagnetic radiation technique," *Process Saf. Environ. Protection*, vol. 119, pp. 285–294, 2018.
[7] F. Derbel, "Performance improvement of fire detectors by means of gas sensors and neural networks," *Fire Saf. J.*, vol. 39, no. 5, pp. 383–398, Jul. 2004.
[8] L. Yuan, R. A. Thomas, J. H. Rowland, and L. Zhou, "Early fire detection for underground diesel fuel storage areas," *Process Saf. Environ. Protection*, vol. 119, pp. 69–74, Oct. 2018.
[9] A. Ayala, B. Fernandes, F. Cruz, D. Macedo, A. L. I. Oliveira, and C. Zanchettin, "KutralNet: A portable deep learning model for fire recognition," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2020, pp. 1–8.
[10] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1251–1258.

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, vol. 25. Stateline, NV, USA, Dec. 2012, pp. 1097–1105.
[12] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.
[13] S. Caelles, K.-K. Maninis, J. Pont-Tuset, L. Leal-Taixe, D. Cremers, and L. Van Gool, "One-shot video object segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 221–230.
[14] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, Feb. 2015.
[15] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1026–1034.
[16] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.
[17] Y. Huang, Y. Cheng, A. Bapna, O. Firat, D. Chen, M. Chen, H. Lee, J. Ngiam, Q. V. Le, and Y. Wu, "GPipe: Efficient training of giant neural networks using pipeline parallelism," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 103–112.
[18] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 6105–6114.
[19] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proc. CVPR*, Jun. 2018, pp. 4510–4520.
[20] O. Kopuklu, N. Kose, A. Gunduz, and G. Rigoll, "Resource efficient 3D convolutional neural networks," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Oct. 2019, pp. 1910–1919.
[21] Y. Li, D. Zhang, and D.-J. Lee, "IIRNet: A lightweight deep neural network using intensely inverted residuals for image recognition," *Image Vis. Comput.*, vol. 92, Dec. 2019, Art. no. 103819.
[22] Y. Chen, H. Fan, B. Xu, Z. Yan, Y. Kalantidis, M. Rohrbach, S. Yan, and J. Feng, "Drop an octave: Reducing spatial redundancy in convolutional neural networks with octave convolution," 2019, *arXiv:1904.05049*.
[23] Q. Xu, Y. Xiao, D. Wang, and B. Luo, "CSA-MSO3DCNN: Multiscale octave 3D CNN with channel and spatial attention for hyperspectral image classification," *Remote Sens.*, vol. 12, no. 1, p. 188, Jan. 2020.
[24] X. Tang, F. Meng, X. Zhang, Y.-M. Cheung, J. Ma, F. Liu, and L. Jiao, "Hyperspectral image classification based on 3-D octave convolution with spatial–spectral attention network," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 3, pp. 2430–2447, Mar. 2021.
[25] K. Vo, T. Le, A. M. Rahmani, N. Dutt, and H. Cao, "An efficient and robust deep learning method with 1-D octave convolution to extract fetal electrocardiogram," *Sensors*, vol. 20, no. 13, p. 3757, Jul. 2020.
[26] A. Jadon, M. Omama, A. Varshney, M. S. Ansari, and R. Sharma, "FireNet: A specialized lightweight fire & smoke detection model for real-time IoT applications," 2019, *arXiv:1905.11922*.
[27] M. T. Cazzolato, L. P. Avalhais, D. Y. Chino, J. S. Ramos, J. A. de Souza, J. F. Rodrigues-Jr., and A. Traina, "FiSmo: A compilation of datasets from emergency situations for fire and smoke analysis," in *Proc. Brazilian Symp. Databases (SBBD)*, 2017, pp. 213–223.
[28] G. Antzoulatos, P. Giannakeris, I. Koulalis, A. Karakostas, S. Vrochidis, and I. Kompatsiaris, "A multi-layer fusion approach for real-time fire severity assessment based on multimedia incidents," in *Proc. 17th Int. Conf. Inf. Syst. Crisis Response Manage. (ISCRAM)*, 2020, pp. 24–27.
[29] Y.-H. Kim, A. Kim, and H.-Y. Jeong, "RGB color model based the fire detection algorithm in video sequences on wireless sensor network," *Int. J. Distrib. Sensor Netw.*, vol. 10, no. 4, Apr. 2014, Art. no. 923609.
[30] N. Grammalidis, E. Çetin, K. Dimitropoulos, F. Tsalakanidou, K. Kose, O. Gunay, B. Gouverneur, D. Torri, E. Kuruoglu, S. Tozzi, A. Benazza, F. Chaabane, B. Kosucu, and C. Ersoy, "A multi-sensor network for the protection of cultural heritage," in *Proc. 19th Eur. Signal Process. Conf.*, Aug. 2011, pp. 889–893.
[31] K. Dimitropoulos, P. Barmpoutis, and N. Grammalidis, "Spatio-temporal flame modeling and dynamic texture analysis for automatic video-based fire detection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 2, pp. 339–351, Feb. 2015.
[32] P. Barmpoutis, K. Dimitropoulos, and N. Grammalidis, "Real time video fire detection using spatio-temporal consistency energy," in *Proc. 10th IEEE Int. Conf. Adv. Video Signal Based Surveill.*, Aug. 2013, pp. 365–370.
[33] H. Wu, D. Wu, and J. Zhao, "An intelligent fire detection approach through cameras based on computer vision methods," *Process Saf. Environ. Protection*, vol. 127, pp. 245–256, Jul. 2019.
[34] J. Sharma, O. Granmo, M. Goodwin, and J. T. Fidje, "Deep convolutional neural networks for fire detection in images," in *Engineering Applications of Neural Networks*. Cham, Switzerland: Springer, 2017, pp. 183–193.
[35] K. Muhammad, J. Ahmad, Z. Lv, P. Bellavista, P. Yang, and S. W. Baik, "Efficient deep CNN-based fire detection and localization in video surveillance applications," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 49, no. 7, pp. 1419–1434, Jul. 2019.
[36] A. Namozov and Y. I. Cho, "An efficient deep learning algorithm for fire and smoke detection with limited data," *Adv. Electr. Comput. Eng.*, vol. 18, no. 4, pp. 121–128, 2018.
[37] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[38] A. Ayala, E. Lima, B. Fernandes, B. L. D. Bezerra, and F. Cruz, "Lightweight and efficient octave convolutional neural network for fire recognition," in *Proc. IEEE Latin Amer. Conf. Comput. Intell. (LA-CCI)*, Nov. 2019, pp. 87–92.
[39] J. Gotthans, T. Gotthans, and R. Marsalek, "Deep convolutional neural network for fire detection," in *Proc. 30th Int. Conf. Radioelektronika (RADIOELEKTRONIKA)*, Apr. 2020, pp. 1–6.
[40] S. H. Oh, S. W. Ghyme, S. K. Jung, and G.-W. Kim, "Early wildfire detection using convolutional neural network," in *Frontiers of Computer Vision*, W. Ohyama and S. K. Jung, Eds. Singapore: Springer, 2020, pp. 18–30.
[41] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size," 2016, *arXiv:1602.07360*.
[42] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.
[43] L. SIfre and S. Mallat, "Rigid-motion scattering for texture classification," 2014, *arXiv:1403.1687*.
[44] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolutional network," *CoRR*, vol. abs/1505.00853, 2015.
[45] Y. Cui, M. Jia, T.-Y. Lin, Y. Song, and S. Belongie, "Class-balanced loss based on effective number of samples," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9268–9277.
[46] L. Studer, M. Alberti, V. Pondenkandath, P. Goktepe, T. Kolonko, A. Fischer, M. Liwicki, and R. Ingold, "A comprehensive study of ImageNet pre-training for historical document image analysis," in *Proc. Int. Conf. Document Anal. Recognit. (ICDAR)*, Sep. 2019, pp. 720–725.
[47] Y. Sari, R. Ramadhan, P. B. Prakoso, and R. A. Pramunendar, "Threshold value optimization to improve fire performance classification using HOG and SVM," in *Proc. Int. Seminar Appl. Technol. Inf. Commun. (iSemantic)*, Sep. 2021, pp. 329–334.
[48] A. R. Singh, S. Athisayamani, S. S. Narayanan, and S. Dhanasekaran, "Fire detection by parallel classification of fire and smoke using convolutional neural network," in *Computational Vision and Bio-Inspired Computing*. Cham, Switzerland: Springer, 2021, pp. 95–105.
[49] A. H. Altowaijri, M. S. Alfaifi, T. A. Alshawi, A. B. Ibrahim, and S. A. Alshebeili, "A privacy-preserving IoT-based fire detector," *IEEE Access*, vol. 9, pp. 51393–51402, 2021.
[50] N. Rahmatov, A. Paul, F. Saeed, and H. Seo, "Realtime fire detection using CNN and search space navigation," *J. Real-Time Image Process.*, vol. 18, no. 4, pp. 1331–1340, Aug. 2021.

ANGEL AYALA received his bachelor's degree in computer engineering from the Universidad Central de Chile, Chile in 2019. He received his master's degree in computer engineering from the Universidade de Pernambuco, Brazil, in 2021. He is currently a PhD's student at the Universidade de Pernambuco. He has already published papers in his early academic career, including efficient deep learning models in the International Joint Conference on Neural Network. His research focuses on controlling fire disasters through autonomous unmanned aerial vehicles commanded via computer vision and reinforcement learning methods.

BRUNO JOSÉ TORRES FERNANDES (Member, IEEE) received the B.Sc., M.Sc., and Ph.D. degrees in computer science from the Federal University of Pernambuco, Recife, Brazil, in 2007, 2009, and 2013, respectively. He is currently an Associate Professor with the University of Pernambuco, where he received the title of Livre-Docente at the end of 2017. He is also a CNPq Productivity Fellow of Technological Development, a Coordinator of the Graduate Program in computer engineering (master's and Ph.D.), UPE, a Coordinator of the Computer Vision Laboratory, Instituto de Inovação Tecnológica (IIT), UPE, and the Head of the Pattern Recognition and Digital Image Processing Research Group, UPE. His research interests include machine learning, computer vision, image processing, and neural networks. He was a recipient of awards, including the 2008 Google Academic Prize as the Top M.Sc. Student with the Federal University of Pernambuco and the Science and Technology Award for Outstanding Research with the Polytechnic School, University of Pernambuco, in 2011 and 2017, respectively.

FRANCISCO CRUZ received a bachelor's degree in engineering and a master's degree in computer engineering from the University of Santiago, Chile, in 2004 and 2006, respectively. In 2017, he received his Ph.D. degree in computer science from the University of Hamburg, Germany, working on cognitive robotics and particularly on interactive reinforcement learning. He was a visiting researcher at the Emergent Robotics Laboratory, Osaka University, Japan in 2015 and the Polytechnic School, University of Pernambuco, Brazil in 2018. Currently, he is a Lecturer in emergent technologies and robotics at the School of IT, Deakin University, Australia. His current research interests include human–robot interaction, cognitive and developmental robotics, interactive and explainable learning, reinforcement learning and affordances, and psychological and bio-inspired models.

DAVID MACÊDO (Graduate Student Member, IEEE) received the B.Sc. degree *(summa cum laude)* in electronic engineering and the M.Sc. degree in computer science from the Universidade Federal de Pernambuco (UFPE), Brazil, where he is currently pursuing the Ph.D. degree in computer science. He was a Visiting Researcher with the Montreal Institute for Learning Algorithms (MILA), Université de Montréal (UdeM), Canada. He co-created and is currently a Collaborator Professor of the deep learning course of the computer science master's and doctorate programs with the Center for Informatics (CIn), UFPE. He is currently a Professor with the Faculdade Nova Roma, Brazil. He has authored approximately 20 deep learning papers published in international peer-reviewed journals and conferences. He is a Reviewer of IEEE journals. His research interests include deep learning, computer vision, natural language processing, speech processing, and trustworthy artificial intelligence.

CLEBER ZANCHETTIN (Member, IEEE) received the D.Sc. degree in computer science from the Center for Informatics, Federal University of Pernambuco, Brazil, in 2008. He is currently an Associate Professor with the Center for Informatics, Federal University of Pernambuco, and a Visiting Professor at Northwestern University, USA. He is also a CNPq Productivity Fellow in technological development. He has authored more than 100 scientific publications in journals and conferences and one edited book. His research interests include pattern recognition, machine learning, deep learning, and image analysis.