KutralNext: An Efficient Multi-label Fire and Smoke Image Recognition Model

Angel Ayala*, David Macêdo‡, Cleber Zanchettin‡, Francisco Cruz† and Bruno Fernandes*
*Escola Politécnica de Pernambuco, Universidade de Pernambuco, Recife, Brasil.
†School of Information Technology, Deakin University, Geelong, Australia.
‡Centro de Informática, Universidade Federal de Pernambuco, Recife, Brasil.
Emails: {aaam, bjtf}@ecomp.poli.br, francisco.cruz@deakin.edu.au, {dlm, cz}@cin.ufpe.br

***Abstract*—Early alert fire and smoke detection systems are crucial for management decision making as daily and security operations. One of the new approaches to the problem is the use of images to perform the detection. Fire and smoke recognition from visual scenes is a demanding task due to the high variance of color and texture. In recent years, several fire-recognition approaches based on deep learning methods have been proposed to overcome this problem. Nevertheless, many developments have been focused on surpassing previous state-of-the-art model's accuracy, regardless of the computational resources needed to execute the model. In this work, is studied the trade-off between accuracy and complexity of the inverted residual block and the octave convolution techniques, which reduces the model's size and computation requirements. The literature suggests that those techniques work well by themselves, and in this research was demonstrated that combined, it achieves a better trade-off. We proposed the KutralNext architecture, an efficient model with reduced number of layers and computacional resources for single- and multi-label fire and smoke recognition tasks. Additionally, a more efficient KutralNext+ model improved with novel techniques, achieved an 84.36% average test accuracy in FireNet, FiSmo, and FiSmoA fire datasets. For the KutralSmoke and FiSmo fire and smoke datasets attained an 81.53% average test accuracy. Furthermore, state-of-the-art fire and smoke recognition model considered, FireDetection, KutralNext uses 59% fewer parameters, and KutralNext+ requires 97% fewer flops and is 4x faster.**

I. Introduction

The[1] presence of fire in some environments is capable of causing massive losses; hence, the early recognition for this kind of accident is primordial. Early recognition of fire can be translated in a quick response to manage the accident, and therefore, high accuracy of fire recognition is also essential. In this regard, a system capable of triggering an alarm with high accuracy is crucial for the response team in charge of monitoring this kind of accident.

Fire accidents can be present in many environments, e.g., open-air, private, or community use spaces, among others, and can be originated because of human intervention, piece of machinery malfunction, unstable state of some structures, or in many other cases as a consequence of other natural disasters. Uncontrolled fire, or blaze, can affect in economic, social, and environmental way principally. This damage could be restored or not. In case it could be restored, considerable effort and consequently, resources are required. A common type of fire accident is the forest fire, which can significantly damage the environment [1] and increase its severity if it spreads.

In Latin America, the forest fires are mainly present in the Amazonia [2] and Chile [3], and have economic and environmental consequences such as mentioned by Urzua et al. [4]. Chile, just in 2014, had more than 8000 fires, which affected 130000ha. After the forest fire, the soil remains damaged [5], and it is difficult for the vegetation to grow again. When this type of accident occurs in the environment, all plants and animal life disappear from the affected zone due to the environment's perturbation. The fires' problem is that they are unpredictable, in the way of when or where they will occur, especially for forest fires. Hence, an early alert system would help to manage these accidents or natural disasters.

This work proposes an efficient deep learning model to recognize fire and smoke as a multi-label classification task specialized for embedded devices such as CCTV devices, mobile and robotic systems. The model's architecture development focused on low computing power devices with high accuracy in acquiring fire and smoke features. In order to obtain a suitable model, different architectures were proposed inspired by generic- and specific-purpose deep learning models and trained with previously used datasets. Hereof, a final efficient architecture was developed after checking different efficient techniques such as convolve methods and convolutional blocks with a specific setup.

II. Fire and Smoke Image Classification

First approaches to fire recognition in computer vision were addressed using techniques based on RGB color space [6], spectral color [7], texture recognition [8], and spatio-temporal treatment [9]. The most recent methods using DL approaches have tackled the problem of fire and smoke recognition through a convolutional neural network (CNN), as a single-label classification task, where the CNN process an input data image by each convolutional layer, reducing its dimensionality into meaningful features. The features acquired by a CNN have been proven to be related to the network's depth. Early layers can obtain simple features like colors and shapes, and final layers process complex features [10]. After rich features were obtained from the input, this data representation is processed by a classifier, which usually is a linear regressor. A few fully connected layers with a considerable amount of hidden

[1] M.Sc. Dissertation

units can also be used as a linear regressor to infer which label corresponds to the image. The most recent methods are detailed as follows.

Sharma et al. [11] developed a custom fire classification model based on VGG16 [10] and ResNet50 [12], two generic-purpose DL models, where the authors just modified the classification stage, adding one fully connected layer at the top of the network implementing transfer learning and fine-tuning methods. Muhammad et al. [13] had proposed a SqueezeNet based-model, where the authors present a custom framework to process the input signal, to classify and locate the fire in a single-label approach. Namozov et al. [14] presented a VGG16 inspired approach with 12 convolutional layers and the adaptative piecewise linear activation [15] function instead of traditional rectified linear units [16]. Their proposal was trained with their dataset with 2440 images labeled as fire and smoke equally balanced. Additionally, the authors implemented data augmentation using Generative Adversarial Networks to create three subsets from the original one.

In terms of specific-purpose models for this task, the following works were found. Gotthans et al. [17] proposed the Fire Detection model to fire and smoke recognition trained with two datasets to compare it against AlexNet [18] and SqueezeNet [19]. The model received an input image of 224x244 pixels with RGB channels, normalized with mean values of (0.485, 0.456, 0.406) and standard deviation of (0.229,0.224, 0.225) for each channel. The authors proposed to achieve a lightweight model capable of recognizing just fire, and fire and smoke in still images. Additionally, they tested the model's execution in the Jatson Nano platform, obtaining the same results. The Fire Detection model reduced in 27% the execution time compared to AlexNet, with only 1% less accuracy. A lightweight model was proposed by Jadon et al. [20], capable of processing images of 64x64 pixels on RGB channels. The architecture comprises three consecutive convolution blocks that contain a convolution layer, an average pooling layer, a dropout layer, and three fully connected layers as the classifier. The proposed approach was focused on being used in an IoT embedded fire alarm system.

III. The KutralNext Proposal

The final contribution achieved in this work evolved from a previously defined architecture model to a specific-purpose model, which improves the size and complexity using novel deep learning techniques. The first exploration approach was a ResNet-based architecture with the octave convolution named OctFiResNet. The second architecture developed was custom-made to recognize fire-only in still images named KutralNet, which presents efficient variations using residual connections, octave convolution, depthwise convolution, and the inverted residual block. That second approach evolved to the final KutralNext model, which improves the fire recognition outcome, including the capability to recognize smoke as well. More details are addressed in the following subsections.

A. ResNet based fire recognition model

The first proposed model is based on the ResNet architecture, named **OctFiResNet** [21], intended to work with the minimum hardware requirements as possible, replacing most of the vanilla convolutions with the octave convolution [22]. The octave convolution processes the signal in two different channels, one for high-frequencies to acquire more detailed features and the other for low-frequencies to more general features. This technique allows the model to work with less memory and fewer flops compared to a vanilla convolution layer. Additional implementation details are in the project's repository[2].

B. Lightweight efficient deep learning model

The following proposal for fire recognition sets a baseline model to develop portable versions focused on reducing the model's complexity in processing the input image. The KutralNet[3] [23] model was developed as a suitable option for limited hardware devices and built other efficient versions using the octave convolution and the inverted residual block to test each efficient technique by themselves and combined. Hereof, three portable models were obtained from this baseline using efficient deep learning techniques. The octave and depthwise convolution [22], [24] demonstrated excellent performance with a sharp reduction of operations and parameters required, resulting in more efficient models.

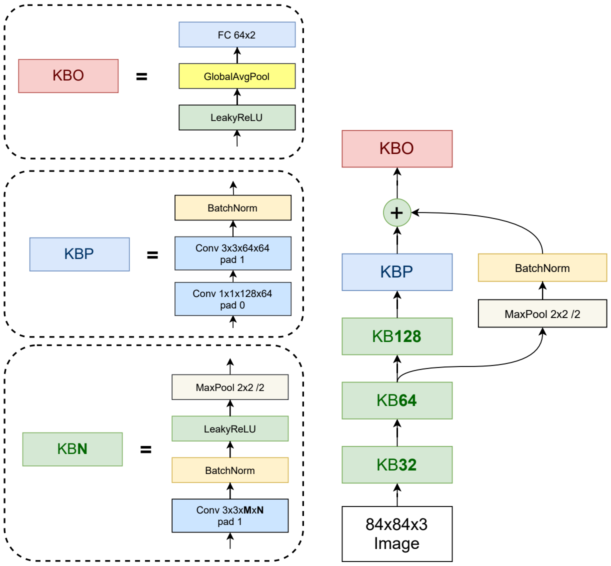
For the case of the separable depthwise convolution in the inverted residual block [24], it increases the number of parameters and reduces the flops efficiently.Given the grouping way to process the convolution channels denoted as $groups = C_{in}$ and $out\_channels = C_{in} * K$, in which the output filters are K times the input filters, reducing the mathematical complexity of the operation. For the octave convolution case, a reduction in both parameters and flops is achieved due to the separate way of processing the filters on high and low frequency, computing the parameters information W into two components $W = [W_H, W_L]$ and exchanging the information between them. Additionally, these convolution techniques, used in different deep learning model architectures, and various tasks such as classification, object detection, and semantic segmentation, achieve a model's size reduction, less computational requirements, and improved performance in some cases. This second proposal combines these techniques, presenting a new convolution type, achieving a valuable trade-off between accuracy, model size, and computational cost. Additional details of the implementations are in the project's repository[4].

1) Baseline model's architecture: The KutralNet model's baseline was inspired by OctFiResNet and FireNet models, mixing between a deep model and a lightweight one, capable of processing 84x84 pixels images in RGB channels. The

[2] OctFiResNet's public repository https://github.com/angel-ayala/fire_recognition

[3] The name took inspiration from Mapuche language or Mapudungun where kütral means fire.

[4] KutralNet's public repository https://github.com/angel-ayala/kutralnet



(a) (b)

Fig. 1. (a) The KutralNet main blocks. The KutralBlockN (KBN) where N refers to the output channels number, KutralBlockP (KBP), and the KutralBlockO (KBO). (b) The baseline KutralNet model with three KBN blocks, a KBP block, a shortcut connection, and a KBO block with two exits.

KutralNet architecture 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 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 comprises two convolution layers and a batch-normalization layer. Finally, the KBO block possesses a LeakyReLU activation, a global average pooling layer, and a fully-connected layer with two exits, one for fire and the other for non-fire labels. This architecture was defined for processing low-dimension images in a lightweight configuration. Each block details are shown in Figure 1a. As shown in Figure 1b, the architecture consists of three KBN blocks, one KBP block, and finally, a KBO output block. A max-pooling and batch-normalization layers, as a shortcut, process the signal from the KB64 block to the final KBO output block. This setup was followed because it has been proved that just a few layers can acquire enough features for a fire classification task to improve the inference time [20]. Additionally, using a shortcut and batch-normalization layers avoids overfitting the model [12]. Also, the LeakyReLU was chosen since a non-zero slope for the negative part improves the results [25] and presents a low-cost implementation.

2) Portable version implementations: The KutralNet portable models development was focused on reducing the model size and computational cost. The octave and depth-wise convolution [22], [24] demonstrated an excellent performance with a sharp reduction of operations and parameters required, resulting in more efficient models.

- **KutralNet Mobile:** Was inspired by MobileNetV2 [24] and presents the implementation of the inverted residual block. In this approach, from the KB64 block, the KutralNet convolution blocks were replaced with the inverted residual block, in which each block contains point-wise and depth-wise convolution with shortcut connections in some cases.
- **KutralNet Octave:** It is based on the KutralNet's architecture, and all the vanilla convolution were replaced with octave convolution with an α parameter of 0.5. Thus, the octave convolution uses the 50% for the *octave feature representation*, which corresponds to the low-frequency channel dealing with global features, and the rest for the high-frequency channel dealing with specific features. Additionally, the octave convolution works using the depth-wise convolution form where it is possible.
- **KutralNet Mobile Octave:** It is the combination of the MobileNetV2 block and the octave convolution. It is the same KutralNet Mobile but replacing the vanilla convolution with the octave convolution combined with depth-wise convolution form. The resultant block can be seen in Figure 2.

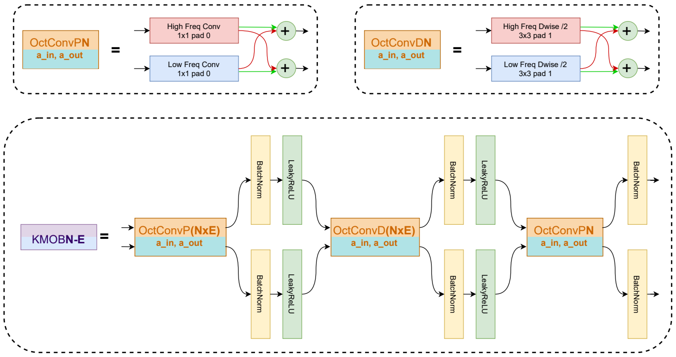


Fig. 2. The KutralNet Mobile Octave main blocks, the KutralMobileOctaveBlockN-E (KMOBN-E), where N refers to the number of output channels and E for the t value of expansion rate. OctConvPN and OctConvDN are the octave convolution version for the separable depthwise convolution layers. This block replaces the KB64, KB128 and KBP blocks of KutralNet with the same number of channels.

C. Multi-label fire and smoke recognition model

All of the previous methods were considered using a single-label fire-flame classification task, indicating if there is a fire presence in the images or not. In this third and final proposal, a multi-label fire and smoke recognition task extends the KutralNet proposal called KutralNext. In terms of architecture, no changes were made in this proposal, and the main changes rely on the classifier exits of the KBO block. For KutralNet, one exit was used for the positive case and the other for the negative case of fire presence, being mutually exclusive. For KutralNext, the first exit indicates fire presence in the image, and the second exit indicates if there is smoke present in the image, being complementary. The models were named KutralNext and KutralNext+, both chosen from KutralNet best models, demonstrating good performance in fire recognition trained from scratch. The models were adjusted for fire and smoke recognition using transfer-learning and Class Balanced loss function, explained later. Experiments have demonstrated that the multi-label approach, in addition to recognizing smoke in the image, it also improves the model's capability to acquire fire's features.

1) 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 to acquire valuable information from a high dimensional input.

For this purpose, we use the ImageNet ILSVRC 2012 dataset [26], which comprises 1.3 million instances with 1,000 classes, designed for a classification and detection competition, being widely used as a models' performance benchmark. Many classical DL models such as ResNet and EfficientNet have been trained with ImageNet and are publicly available in different repositories to be used by the community. We use the ImageNet dataset to training the baseline, and the efficient architectures for later use in the fire and smoke classification task.

2) 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 under-represented classes. A novel framework is implemented in our proposal to deal with this class imbalance issue, which 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 [27], and 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

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

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) [28], which 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

$$\text{FL}(p_y) = -(1-p_y)^\gamma \log(p_y), \tag{2}$$

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

$$p_y = \frac{1}{(1+\exp -z_y)}.$$

TABLE I
QUANTITY OF IMAGES PER CLASS PRESENT IN EACH DATASET.

Dataset	Set	F & S	Fire	Smoke	None	Total
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		2,341	2,442	1,386	6,675	12,844

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

$$\text{CB}_{\text{focal}}(z,y) = -\frac{1-\beta}{1-\beta^{n_y}}(1-p_y)^\gamma \log(p_y), \tag{3}$$

where z is the model's predicted class probability.

IV. Experimental Results

Three publicly available datasets were used to benchmark the performance of the KutralNext and KutralNext+ against the FireDetection and FireNet models. The datasets were designed for a fire or smoke single-label classification task, with fire, smoke, or none classes, named FiSmo[5] [29], FireNet[6] [20], and FireSmoke[7]. All the datasets were previously used in fire and fire and smoke classification tasks as presented in [17], [20], [30]. For this project, 12,844 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. The FireNet and FireSmoke datasets were merged into a new one called KutralSmoke with $6,296$ images, and a test subset with $1,171$ images. 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. More details are presented in Table I, where the F & S column is used to represent fire and smoke class.

Overall all the models were compared using the validation and testing accuracy, the Receiver Operating Characteristic (ROC) curve, the area under the ROC curve, the floating-point operations (flops), the number of parameters, and the time required to process the entire test dataset used for each experiment. Those metrics were selected to compare each model generalization and acquisition of the fire and smoke features under the same and different data distributions. Each used model is represented in Table II by the computational cost in terms of flops and parameters.

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

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

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

TABLE II
THE COMPUTATIONAL COST OF EACH MODEL USED IN THIS WORK REPRESENTED WITH FLOPS AND PARAMETERS ORDERED BY PARAMETERS NUMBER.

Model$_{InputSize}$	**Flops**	**Parameters**
KutralNext$_{84x84}$	76.85M	138.91K
KutralNext+$_{84x84}$	24.59M	185.25K
FireDetection$_{224x224}$ [17]	783.50M	335.53K
FireNet$_{64x64}$ [20]	8.94M	646.82K
OctFiResNet$_{96x96}$	928.95M	956.23K

A. Multi-label classification: Fire and smoke recognition

In this experiment, we check out the performance in the fire and smoke multi-label recognition task of our models' proposals with two datasets used for training and one dataset for testing. The training datasets were FiSmo and KutralSmoke, and the testing dataset was KutralSmoke Test. With those datasets, the models' were trained and compared with different data distribution of the corresponding labels and checking its generalization. This fire and smoke classification task analyzed each label separately under a multi-label approach due to the chance of appearing the fire or smoke class in the same image. The Table III shown the statistics results of each model trained over all the datasets with averaged values for the test accuracy and test time and Table IV presented each model's test performance. Our proposals are the best in recognize fire and smoke, being the most time inexpensive models. The classification was considered binary, considering fire, smoke, or both classes as a true label and none class as a false label.

The Table III shown the models' training performance, where it can be observed that in average testing accuracy, KutralNext+ performs the best with the same and different data distribution obtaining an 81.53%, being better than KutralNext with 79.03% an all previous models. Now, in terms of time required to process the 1,171 testing images, OctFiResNet was the most time-consuming, taking over 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 which requires less time with 0.41 seconds. FireNet is the model that requires less time to process the images; nevertheless, it also presents the lowest mean test accuracy. In this regard, KutralNext+ surpasses the state-of-the-art fire recognition models, requiring less time in processing the test data images.

A general overview of each model in terms of AUROC and precision in the test dataset is shown in Table IV. In the first place, for the fire label, KutralNext has shown 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%, taking the first place, followed by KutralNext+ with 93.40%. In overall, all the models present a good performance to detect fire in this approach. However, for the smoke label, a lower outcome has been shown in AUROC and precision terms. KutralNext+ achieved a remarkable AUROC value with 89.59% and precision of 56.27%, followed by KutralNext with 87.00% and 46.92%, respectively, being the best model in comparison with previous extended models to acquire smoke features and recognize it under a multi-label approach. All of the models have been shown better outcomes trained over the same data distribution than a different data distribution.

TABLE III
KUTRALNEXT TRAINING RESULTS DURING 5 EXECUTIONS IN THE FIRE AND SMOKE RECOGNITION TASK.

DS	**Model**	**Test acc.**	**Test (ms)**
KutralSmoke	FireDetection [17]	77.59% ± 3.22%	1883 ± 81
	FireNet [20]	77.11% ± 3.60%	**339 ± 23**
	KutralNext	86.70% ± 2.02%	430 ± 21
	KutralNext+	**88.08% ± 0.69%**	603 ± 34
	OctFiResNet	79.03% ± 4.58%	2040 ± 11
FiSmo	FireDetection [17]	63.04% ± 8.60%	1856 ± 99
	FireNet [20]	56.89% ± 6.26%	**335 ± 22**
	KutralNext	71.36% ± 2.31%	424 ± 21
	KutralNext+	**74.98% ± 3.22%**	624 ± 33
	OctFiResNet	56.69% ± 2.38%	2046 ± 6
Average	FireDetection [17]	70.32% ± 5.91%	1870 ± 90
	FireNet [20]	67.00% ± 4.93%	**337 ± 23**
	KutralNext	79.03% ± 2.16%	427 ± 21
	KutralNext+	**81.53% ± 1.96%**	614 ± 33
	OctFiResNet	67.86% ± 3.48%	2043 ± 9

Figure 3 and Figure 4 shows the mean ROC values obtained for the models trained over all the datasets to compare each models' performance in terms of feature acquisition for each model. The KutralNext proposals presented the best results for both classes from the used datasets, capable of acquiring features at a low false-positive rate. Remarkable results were obtained for the smoke label compared with previous models, as shown in Figure 4a and Figure 4b. Additionally, KutralNext and KutralNext+ obtained the best results under a different data distribution as the case for the FiSmo dataset. In this way, their implemented techniques efficiency has been demonstrated because the models' design was not meant to recognize smoke. Even so, it achieved the best results in smoke class features.

V. CONCLUSION

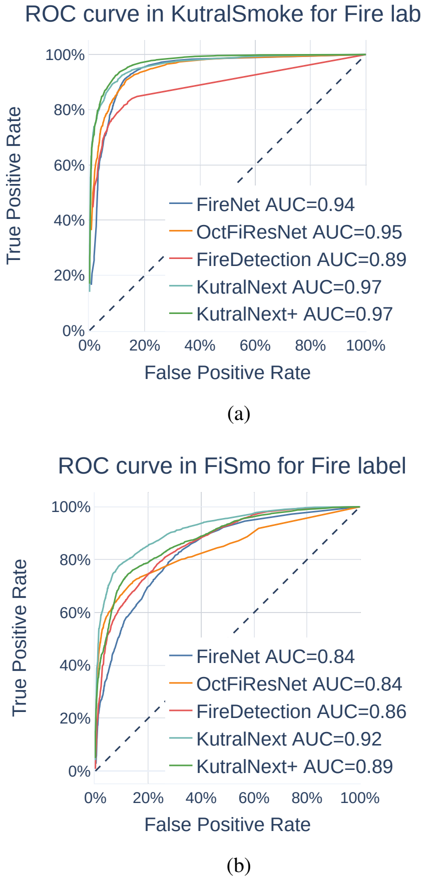
Fire disasters may lead to massive losses affecting in an environmental, social, and economic way, caused by natural or human causes. Hereof, an early detection system is affordable to manage this kind of accident, reducing the blazes' affected area. Ensure a real-time processing algorithm with high accuracy is challenging for deep learning due to previous state-of-the-art generic-purpose models that are mathematically complex designed to recognize a higher amount of classes. In this regard, a deep learning model for fire and smoke recognition under an efficient and reduced size scope was developed in this work.

For this work purpose, a study about the current fire and smoke recognition algorithm and novel techniques that opti-

TABLE IV
KUTRALNEXT PERFORMANCE DURING 5 EXECUTIONS IN FIRE AND SMOKE RECOGNITION TASK.

DS	Model	AUROC	Precision
		Fire Label	
KutralSmoke	FireDetection [17]	88.72% ± 4.04%	95.02% ± 1.87%
	FireNet [20]	94.18% ± 1.68%	94.07% ± 1.28%
	KutralNext	96.96% ± 0.49%	**97.12% ± 0.80%**
	KutralNext+	**97.46% ± 0.43%**	96.69% ± 1.21%
	OctFiResNet	94.84% ± 2.67%	94.74% ± 2.11%
FiSmo	FireDetection [17]	85.73% ± 7.74%	92.61% ± 4.77%
	FireNet [20]	83.66% ± 5.49%	90.74% ± 3.48%
	KutralNext	**91.98% ± 2.97%**	93.64% ± 4.11%
	KutralNext+	89.35% ± 2.03%	91.74% ± 3.48%
	OctFiResNet	84.25% ± 3.61%	**96.70% ± 2.03%**
Average	FireDetection [17]	87.23% ± 5.89%	93.82% ± 3.32%
	FireNet [20]	88.92% ± 3.59%	92.40% ± 2.38%
	KutralNext	**94.47% ± 1.73%**	95.38% ± 2.45%
	KutralNext+	93.40% ± 1.23%	94.22% ± 2.35%
	OctFiResNet	89.54% ± 3.14%	**95.72% ± 2.07%**
		Smoke Label	
KutralSmoke	FireDetection [17]	70.78% ± 3.84%	29.30% ± 2.59%
	FireNet [20]	72.22% ± 1.55%	28.00% ± 1.73%
	KutralNext	91.74% ± 1.23%	**52.91% ± 3.82%**
	KutralNext+	**92.59% ± 1.77%**	52.19% ± 6.55%
	OctFiResNet	76.42% ± 6.25%	31.49% ± 5.90%
FiSmo	FireDetection [17]	67.38% ± 3.92%	33.06% ± 3.21%
	FireNet [20]	67.79% ± 5.35%	35.01% ± 6.30%
	KutralNext	82.27% ± 1.19%	40.93% ± 2.41%
	KutralNext+	**86.59% ± 3.22%**	**60.35% ± 12.66%**
	OctFiResNet	66.95% ± 4.95%	29.75% ± 3.68%
Average	FireDetection [17]	69.08% ± 3.88%	31.18% ± 2.90%
	FireNet [20]	70.00% ± 3.45%	31.51% ± 4.02%
	KutralNext	87.00% ± 1.21%	46.92% ± 3.11%
	KutralNext+	**89.59% ± 2.50%**	**56.27% ± 9.60%**
	OctFiResNet	71.69% ± 5.60%	30.62% ± 4.79%

mize the models' performance, such as octave and separable depthwise convolution and the inverted residual block. OctFiResNet performed better results than previous approaches using FiSmo and FireNet datasets. In this way, it demonstrated that FiSmo and FireNet datasets are suitable options to be used in a fire and smoke classification task given the challenging fire scenarios and the smoke labeled images. Those datasets allowed the fire recognition model's main architecture development, which fits the defined scope of reduced size and computational cost for this work. The KutralNet proposal was able to perform better than previous models, proving the effectiveness in recognizing fire. Furthermore, it was optimized with novel deep learning convolution methods, with KutralNet Mobile Octave as the best portable model in this task. A final proposal with those models was completed, named KutralNext and KutralNext+. Both models were trained under more complex representation data features in the ImageNet dataset to be optimized with the fire and smoke labeled datasets. A novel approach for fire and smoke recognition was proposed with 138.9K parameters, and 76.9M flops, with an efficient model developed in this work. 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, being 16% more accurate during testing in the fire and smoke recognition than FireDetection. Hence, it is executed 4x faster with better generalization.



(a)

(b)

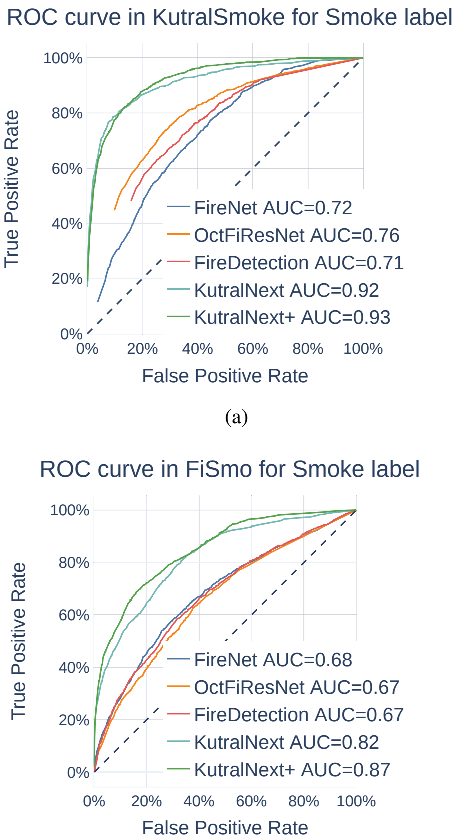
Fig. 3. Multi-label fire class ROC curve mean performance for each model trained over different data distribution datasets. (a) and (b) present the models' performance trained over the KutralSmoke and FiSmo dataset respectively. In (a) the KutralNext+ model achieves the best performance under low false positive rate. In (b) the KutralNext and KutralNext Mobile models, respectively achieves the best performance at low false positive rate.

ACKNOWLEDGMENT

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001, Fundação de Amparo a Ciência e Tecnologia do Estado de Pernambuco (FACEPE), and Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) - Brazilian research agencies.

REFERENCES

[1] J. Moreno, M. Arianoutsou, A. González-Cabán, F. Mouillot, W. Oechel, D. Spano, K. Thonicke, V. Vallejo, and R. ed.) Vélez, "Forest fires under climate, social and economic changes in europe, the mediterranean and other fire-affected areas of the world," *FUME : lessons learned and outlook*, 1 2014.



(a)

(b)

Fig. 4. Multi-label smoke class ROC curve mean performance for each model trained over different data distribution datasets. (a) and (b) present the models' performance trained over the KutralSmoke and FiSmo dataset respectively. (b) the KutralNext+ model achieves the best performance under low false positive rate. (d) the KutralNext and KutralNext Mobile models, respectively achieves the best performance at low false positive rate.

[2] J. Barlow and C. A. Peres, "Avifaunal responses to single and recurrent wildfires in amazonian forests," *Ecological Applications*, vol. 14, no. 5, pp. 1358–1373, 2004.

[3] X. Úbeda and P. Sarricolea, "Wildfires in chile: A review," *Global and Planetary Change*, vol. 146, pp. 152 – 161, 2016.

[4] N. V. Urzúa Valenzuela and M. F. Cáceres Bueno, "Incendios forestales: principales consecuencias económicas y ambientales en chile," *Revista Interamericana de Ambiente y Turismo - RIAT*, vol. 7, no. 1, pp. 18 – 24, 2011.

[5] V. Quintanilla PÉrez, "Perturbaciones a la vegetación nativa por grandes fuegos de 50 años atrás, en bosques nordpatagónicos. caso de estudio en chile meridional," *Anales de Geografía de la Universidad Complutense*, vol. 28, no. 1, pp. 85 – 104, 5 2008.

[6] Y.-H. Kim, A. Kim, and H.-Y. Jeong, "Rgb color model based the fire detection algorithm in video sequences on wireless sensor network," *International Journal of Distributed Sensor Networks*, vol. 10, no. 4, p. 923609, 2014.

[7] G. Nikos, A. Cetin, K. Dimitropoulos, F. Tsalakanidou, K. Kose, O. Gunay, B. Gouverneur, D. Torri, E. Kuruoglu, S. Tozzi, A. Benazza-Benyahia, F. Chaabane, B. Kosucu, and C. Ersoy, "A multi-sensor network for the protection of cultural heritage," 8 2011.

[8] K. Dimitropoulos, P. Barmpoutis, and N. Grammalidis, "Spatio-temporal flame modeling and dynamic texture analysis for automatic video-based fire detection," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 2, pp. 339–351, 2 2015.

[9] P. Barmpoutis, K. Dimitropoulos, and N. Grammalidis, "Real time video fire detection using spatio-temporal consistency energy," in *2013 10th IEEE International Conference on Advanced Video and Signal Based Surveillance*, 8 2013, pp. 365–370.

[10] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.

[11] 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: Springer International Publishing, 2017, pp. 183–193.

[12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015.

[13] 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 Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 7, pp. 1419–1434, 2018.

[14] A. Namozov and Y. I. Cho, "An efficient deep learning algorithm for fire and smoke detection with limited data," *Advances in Electrical and Computer Engineering*, vol. 18, pp. 121–128, 2018.

[15] F. Agostinelli, M. Hoffman, P. Sadowski, and P. Baldi, "Learning activation functions to improve deep neural networks," *arXiv preprint arXiv:1412.6830*, 2014.

[16] A. F. Agarap, "Deep learning using rectified linear units (relu)," *arXiv preprint arXiv:1803.08375*, 2018.

[17] J. Gotthans, T. Gotthans, and R. Marsalek, "Deep convolutional neural network for fire detection," in *2020 30th International Conference Radioelektronika (RADIOELEKTRONIKA)*, 2020, pp. 1–6.

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds., vol. 25. Curran Associates, Inc., 2012, pp. 1097–1105.

[19] 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¡ 0.5 mb model size," *arXiv preprint arXiv:1602.07360*, 2016.

[20] 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," *CoRR*, vol. abs/1905.11922, 2019.

[21] A. Ayala, E. Lima, B. Fernandes, B. L. D. Bezerra, and F. Cruz, "Lightweight and efficient octave convolutional neural network for fire recognition," in *2019 IEEE Latin American Conference on Computational Intelligence (LA-CCI)*, 2019, pp. 87–92.

[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," *arXiv preprint arXiv:1904.05049*, 2019.

[23] A. Ayala, B. Fernandes, F. Cruz, D. Macêdo, A. L. I. Oliveira, and C. Zanchettin, "Kutralnet: A portable deep learning model for fire recognition," in *2020 International Joint Conference on Neural Networks (IJCNN)*, 2020, pp. 1–8.

[24] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2018, pp. 4510–4520.

[25] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolutional network," *CoRR*, vol. abs/1505.00853, 2015.

[26] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[27] Y. Cui, M. Jia, T.-Y. Lin, Y. Song, and S. Belongie, "Class-balanced loss based on effective number of samples," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9268–9277.

[28] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.

[29] 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 *Brazilian Symposium on Databases - SBBD*, 2017, pp. 213–223.

[30] 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," Mar. 2020.