KUTRALNE UN MODELO PORTÁTIL DE APRENDIZAJE PROFUNDO PARA RECONOCIMIENTO DE FUEGO

Un modelo eficiente para el reconocimiento de fuego en sistemas de visión portátiles.

• Ángel Ayala

Estudiante de Magíster en la Universidade de Pernambuco, Brasil

Bruno Fernándes

Profesor adjunto de la Universidade de Pernambuco, Brasil

 Francisco Cruz Profesor de la Escuela de Ingeniería de la

Universidad Central de Chile, Chile

 David Macêdo Profesor adjunto de la Universidade Federal de Pernambuco, Brasil

• Adriano L. I. Oliveira Profesor adjunto de la Universidade Federal de Pernambuco, Brasil

Cleber Zanchettin

Profesor adjunto de la Universidade Federal de Pernambuco, Brasil

dizaje profundo para el recono-cimiento de fuego, es un enfoque prometedor ya que no requiere de sensores específicos y puede ser fátes dispositivos. Sin embargo, estos métodos presentan un alto costo este trabajo, proponemos una nueva do que requiere menos operaciones de punto flotante (flops) para el re-conocimiento del fuego. Proponemos además, un enfoque portátil median-te el uso de técnicas modernas como ciones "octave" y "depthwise", que reducen la complejidad del modelo. para su implementación en un dispositivo móvil.

El enfoque de aprendizaje profundo (DL por su nombre en inglés) [1] ha demostrado ser adecuado para automatizar la adquisición de características a partir de datos complejos en tareas de aprendizaje automático. De esta manera, el DL trabaja en múltiples niveles de abstracción para la representación de datos. Teniendo en cuenta esto, el uso de visión por computador para el reconocimiento de fuego reduce la necesidad de sensores específicos, siendo adecuado para la implementación en dispositivos portátiles, remotos y móviles. Sin embargo, los enfoques de la DL presentan algunos desafíos: i) los recursos computacionales necesarios; ii) la complejidad en los cálculos y el tamaño del modelo; y iii) la cantidad de datos necesarios para su entrenamiento, entre otros. Los desafíos mencionados anteriormente pueden abordarse en el desarrollo de modelos para dispositivos móviles [2], que ha sido un área menos explorada en la literatura sobre DL. Para que los sistemas autónomos o robóticos sean portátiles, deben ser trabajar con una potencia de procesamiento, almacenamiento y energía limitados, además de proporcionar un eficiente tiempo de inferencia.

Constructor tab.data-api"

tion, this)).on("click.bs.affix.data-api

vis.checkPosition()};c.VERSION=

Algunos trabajos han abordado modelos portátiles de aprendizaje profundo utilizando formas eficientes de procesar una convolución, como los presentados por Sandler et al. [3]. Los autores propusieron el "inverted residual block" (véase la Figura 1(a)) con convoluciones "pointwise" y "depthwise" para simplificar el procesamiento de la señal de entrada. Así, el modelo ofrece un interesante balance entre precisión, número de operaciones y número de parámetros en consideración con otros modelos como ShuffleNet [4] y NasNet [5]. Otro diseño eficiente para modelos portátiles, es la convolución "octave" presentada por Chen et al. [6] (ver Figura 1(b)), que disminuye la redundancia espacial, separando el procesamiento en una convolución de frecuencia espacial alta y baja. En este caso, el tamaño de la separación depende de un parámetro a, que define la representación de la "octava característica" factorizando el mapeamento de características en grupos para cada frecuencia.





(a) Inverted residual block: Las capas sombreadas diagonalmente no usan no-linearidades. El grosor de cada bloque indica el número relativo de canales. Los residuos invertidos conectan los cuellos de botella. Adaptado de Ref.[3].

(b) Convolución octave: Las flechas verdes actualizan la información, mientras que las flechas rojas facilitan el intercambio de información entre ambas frecuencias. Adaptado de Ref.[6].

Figura 1

Bloques de convolución esenciales usados en este trabajo.

ansitionEnd c,a.fn.tab.noConflict

this.each(function(){var d=a(this),e=d.data("bs.aff n(b,d){this.options=a.extend({},c.DEFAULTS,d),this En este documento proponemos un nuevo modelo de aprendizaje profundo para el reconocimiento de fuego denominado KutralNet, que posee cinco capas y requiere un 92% menos de operaciones de punto flotante (flops) para su procesamiento en comparación con los enfoques anteriores. Este modelo se utiliza como base para construir modelos portátiles que comparan la eficiencia en el procesamiento de la señal de los métodos de inverted residual block, la convolución depthwise y octave.

Nuestro mejor modelo, *KutralNet Mobile Octave, presenta una competitiva precisión en validación y alto rendimiento AUROC a pesar de utilizar un 71% menos de parámetros* en comparación con FireNet, comparados con otros enfoques para el reconocimiento de fuego en los conjuntos de datos FireNet y FiSmo.

Desarrollo

KutralNet tiene por objeto reducir la complejidad de un modelo de aprendizaje profundo para procesar una imagen y decidir si tiene o no presencia de fuego. Esta propuesta establece un modelo base para desarrollar modelos portátiles que se adapten para su uso en dispositivos con hardware limitado.

A. Arquitectura del modelo base

La base para KutralNet está inspirada en los modelos FireNet, OctFiResNet y un modelo modificado de ResNet50. KutralNet es el resultado de la mezcla entre un modelo profundo y uno ligero, capaz de procesar imágenes de 84x84 píxeles en canales RGB. El modelo puede verse en la Figura 2.



Figura 2

La arquitectura KutralNet funciona con imágenes de 84x84 píxeles en canales RGB. Para el clasificador, una capa Global average pooling entrega las características a la capa totalmente conectada (FC) con dos salidas, una para la etiqueta de fuego y otra para la etiqueta de no fuego. Consecuentemente, se implementa una función softmax como activación en la parte superior de la red.

Los tres primeros bloques consisten en una capa de convolución sin sesgo (o bias) con filtros (o kernel) de 3x3, seguida de una capa de normalización por lotes, continuando con una activación LeakyReLU, y finalmente, una capa Max-pooling de 2x2 con stride igual a 2. Cuando la señal pasa de un bloque al otro, aumenta el número de filtros y reduce su dimensión. En el último bloque, hay dos capas de convolución y una capa de normalización por lotes. Una conexión directa une desde el segundo bloque con el bloque final de convolución, procesando la señal con una capa Max-pooling de 2x2 con stride 2 y una capa de normalización por lotes. En la parte superior de la capa posterior al salto, la señal pasa a través de una activación LeakyReLU y una capa de Global average pooling al clasificador, que consiste en una capa totalmente conectada con dos neuronas en la salida. Esta arquitectura está definida para procesar imágenes de baja dimensión con una configuración ligera. Se ha demostrado que pocas capas pueden

adquirir suficientes características para reconocer fuego con el fin de optimizar el tiempo de inferencia [7]. Además, el uso de conexiones directas y capas de normalización por lotes evita el sobre aprendizaje en el modelo [8]. Hemos elegido LeakyReLU ya que una pendiente distinta de cero para la parte negativa de la función, mejora los resultados [9] y es implementada a bajo costo. El resumen de los modelos con los números de los parámetros y las operaciones necesarias para el procesamiento de imágenes se encuentra en la Tabla I. El valor de los flops del modelo FireNet no se presenta debido a la inestabilidad durante la medición, que aumenta el valor en cada ejecución.

Modelo (Tamaño de entrada)	Parámetros	Flops
ResNet50 (224x224)	31.91M	4.13G
OctFiResNet (96x96)	956.23K	928.95M
FireNet (64x64)	646.82K	-
KutalNet (84x84)	138.91K	76.85M

Tabla I

El costo computacional de cada modelo implementado representado como parámetros y flops.

B. Implementación de versiones portátiles

El modelo KutralNet es la base utilizada para desarrollar modelos portátiles, centrándose en la reducción del tamaño y el costo computacional del modelo. La convolución octave Ref.[6] y depthwise [3], muestran un excelente rendimiento con una gran reducción de operaciones y parámetros requeridos, resultando en modelos eficientes. Esta reducción es el resultado de convoluciones con filtros de baja dimensión en ambos casos. Para la convolución depthwise, los canales de filtros se procesan en grupos, donde grupos = $C^{entrada}$ y C^{salida} = $C^{entrada}$ K en que los canales de salida son K veces los canales de entrada, reduciendo así la complejidad matemática en la operación. En el caso de la convolución octave, la señal se procesa de forma separada en filtros de alta y baja frecuencia calculando los parámetros W en dos componentes W=[W_H,W_L] para posteriormente, intercambiar la información entre ellos. Además, estas técnicas de convolución, han sido utilizadas en diferentes arquitecturas de modelos de aprendizaje profundo, y en diversas tareas como clasificación, detección de objetos y segmentación semántica, consiguiendo una reducción del tamaño del modelo, menos requisitos de cálculo y un rendimiento ligeramente mayor en algunos casos. Esto es útil

para nuestro propósito, y presentamos un nuevo tipo de convolución que combina la convolución depthwise con sus operaciones de filtro de grupo y la convolución octave, que logra un valioso equilibrio entre precisión, tamaño del modelo y costo computacional. Nuestras diferentes versiones portátiles implementadas son las siguientes:

> **KutralNet Mobile**: Inspirada en MobileNetV2 [3], implementa el inverted residual block. En este enfoque, a partir del segundo bloque de KutralNet, los bloques de convolución fueron reemplazados por el inverted residual block, en el que cada bloque contiene una convolución pointwise y depthwise con conexiones directas en algunos casos.

> **KutralNet Octave**: Basada en la arquitectura de KutralNet, reemplaza todas las convoluciones regulares por convoluciones octave con un parámetro α de 0,5. Así, la convolución octave utiliza el 50% para representar las características octave, que corresponde al canal de baja frecuencia y se encarga de las características generales, y el resto para el canal de alta frecuencia que se encarga de características específicas.

> **KutralNet Mobile Octave**: Es el resultado de la combinación entre inverted residual block y la convolución octave. Es el mismo KutralNet Mobile pero sustituye las convoluciones regulares por la convolución octave con el método de convolución depthwise. El bloque resultante puede verse en la Figura 3.

Todos los modelos portátiles presentan el mismo clasificador en la parte superior de la red, compuesto de una activación LeakyReLU, que pasa a través de una Global average pool y una capa totalmente conectada con dos neuronas en la salida. Se pueden ver detalles adicionales de las implementaciones en el repositorio del proyecto. En la Tabla II se puede ver un resumen de los parámetros y operaciones de los modelos KutralNet. El número de operaciones de punto flotante (flops) y el número de parámetros de un modelo son las métricas definidas para medir los requisitos del modelo para el procesamiento de imágenes y almacenamiento del mismo, respectivamente. Cuantos menos parámetros, menos tamaño de disco se requiere. Además, cuanto menor sea el número de flops, menor será el costo computacional. Dado que nuestro trabajo se centra en el desarrollo de un modelo de aprendizaje profundo móvil, cuanto menos valor tengan ambas métricas, más adecuado será el modelo para este fin. Para el caso de las variaciones de Mobile y Mobile Octave, los modelos presentan un mayor número de parámetros pero con una alta reducción de los flops respecto al modelo base.

Modelo (Tamaño de entrada)	Parametros	Flops
KutralNet (84x84)	138.91K	76.85M
KutralNet Mobile (84x84)	173.09K	43.27M
KutralNet Octave (84x84)	125.73K	29.98M
KutalNet Mobile Octave (84x84)	185.25K	24.59M

Tabla II

El costo computacional de cada variación portátil de kutralnet representada como parámetros y flops.

C. Conjuntos de datos

Para el entrenamiento, validación y prueba de los modelos, se utilizaron dos conjuntos de datos en este trabajo. El primero se llama FireNet como el modelo Ref.[7] y contiene subconjuntos de entrenamiento y prueba, con 2425 y 871 imágenes, respectivamente. El segundo es el conjunto de datos FiSmo propuesto por Cazzolato et al. [10], que ha sido recientemente publicado con un total de 6063 imágenes. Además, hemos utilizado un subconjunto de FiSmo compuesto de 1968 imágenes igualmente balanceadas entre la etiqueta de fuego y la de no fuego. También se utiliza una versión aumentada de FiSmo, agregando 485 imágenes negras etiquetadas como no-fire, para comprobar la respuesta de los modelos a este tipo de aumento. Además de la versión equilibrada de FiSmo, también hemos utilizado una versión aumentada de este subconjunto, que sustituye 98 imágenes no-fire por imágenes negras. Más detalles de las imágenes del conjunto de datos se encuentran en la Tabla III, donde FiSmoA es la versión aumentada de FiSmo, FiSmoB es la versión equilibrada de FiSmo, y FiSmoBA es la versión aumentada equilibrada de FiSmo.



Figura 3

Bloque de KutralNet Mobile Octave. La parte más a la izquierda y derecha del bloques presentan una convolución pointwise y, en el centro, una convolución depthwise, todo en combinación con la convolución octave con $\alpha = 0,5$.

Dataset	Fire	No Fire	Total
FireNet (training)	1124	1302	2425
FireNet (testing)	593	278	871
FiSmo	2004	4059	6063
FiSmo A	2004	4544	6548
FiSmo B	984	984	1968
FiSmo BA	984	984	1968

Tabla III

Cantidad de imágenes por etiqueta presentes en cada conjunto de datos usado para este trabajo.

Análisis

Las dos siguientes subsecciones separan los experimentos de aprendizaje profundo para el reconocimiento del fuego. El primer experimento fue con el modelo base propuesto, KutralNet, que utiliza novedosas técnicas de aprendizaje profundo para la clasificación de imágenes. Los resultados comparan KutralNet con otros modelos de aprendizaje profundo presentados anteriormente. Posteriormente, se optimizó el costo computacional del modelo, explorando los beneficios de los diferentes enfoques portátiles presentados en los últimos años como inverted residual block, convolución depthwise y octave, obteniendo una precisión similar al del modelo base.

A. Comparación del modelo base

La comparación del modelo base se realiza con otros tres modelos, evidenciando la eficiencia y la ligereza de nuestra propuesta. El primer modelo es FireNet de Jadon et al. [7], que posee unas pocas capas de convolución y forma parte de un sistema de alarma de incendios. El segundo modelo es presentado por Sharma et al. [11], donde ResNet50 pre-entrenado hace la extracción de características para un clasificador perceptrón multicapa con 4096 unidades ocultas. Por último, OctFiResNet [12], una versión reducida de ResNet50 que presenta menos capas y reemplaza la gran mayoría de convolución regular por convolución octave.

Una primera comparación se realizó con el dataset FireNet con 2425 imágenes, 1124 con la etiqueta de fuego, y 1301 con la etiqueta de no-fuego. Los primeros resultados muestran una precisión de validación del 93,83%; 96,02%; 95,34% y 98,22% para FireNet, KutralNet, OctFiResNet y ResNet50 respectivamente para el dataset FireNet. Adicionalmente, los resultados de precisión de prueba son 88,98%; 83,70%; 88,18% y 89,44% para FireNet, KutralNet, OctFiResNet y ResNet50. Para comprobar la capacidad de generalización de los modelos, para el entrenamiento y la validación, se utilizó el dataset FiSmo, para posteriormente ser probados con el dataset FireNet-Test, como era de esperar, obtienen una menor precisión para el entrenamiento con el dataset FiSmo.

A continuación, se evaluó la predicción del modelo con una imagen negra como entrada. Todos los modelos entrenados con el dataset FireNet clasificaron erróneamente la imagen negra, ocurriendo lo mismo con los modelos entrenados con FiSmo, clasificando erróneamente la imagen negra, presentando algunas excepciones. El conjunto de datos FiSmo fue aumentado, para tratar este problema de clasificación errónea, añadiendo un 10% de las imágenes etiquetadas de no-fuego, con imágenes negras. El aumento para esta tarea mostró ser útil en la etapa de entrenamiento y de prueba; ambos resultados pueden observarse en la Figura 4(a) y en la Figura 4(b). Los modelos más profundos superan los resultados del modelo FireNet.



(a) Precisión de validación obtenida para cada dataset por los diferentes modelos. ResNet50 funciona mejor con la versión aumentada de FiSmo y para el dataset FireNet, seguido de Kutral-Net. Además, Kutralnet tiene mejor desempeño con FiSmo.



(b) Precisión de prueba para cada modelo entrenado con un dataset diferente y probado con FireNet-Test. El modelo KutralNet es un 5,7% más bajo que ResNet50 con FireNet, y obtiene la mejor precisión entrenado sobre FiSmo. ResNet50 obtiene un mejor rendimiento con FiSmoA, seguido por KutralNet.

Figura 4

Resultados de entrenamiento de los conjuntos de datos FireNet, FiSmo, incluyendo FiSmoA, la versión aumentada del FiSmo (con 485 imágenes negras adicionales). Los conjuntos de datos se dividieron en 70/30 para FireNet y 80/20 para ambas variantes de FiSmo. El aumento con imágenes negras permite una mejor generalización en todos los modelos, pero sólo una ligera diferencia para KutralNet.

El rendimiento de las pruebas de los modelos entrenados con diferentes conjuntos de datos se muestran en la Figura 5. Como puede verse, comparando la Figura 5(b) con la 5(c), las imágenes negras añadidas al conjunto de datos FiSmo muestran una reacción en el comportamiento de la curva ROC. En el caso de los modelos KutralNet y ResNet50 se observa una mejora, mientras que FireNet muestra una disminución, y con OctFiResNet se mantiene casi igual. En cuanto al índice AUROC, FireNet logra un mejor valor en todos los conjuntos de datos, pero como se presenta en la Figura 4(b) para FiSmo y FiSmoA logra una baja precisión de prueba. Para visualizar la comparación de los modelos de una manera más directa, la Tabla IV muestra el valor medio de precisión de prueba y el índice AUROC para todos los conjuntos de datos y modelos. Nuestra propuesta presenta una buena precisión para validación y prueba utilizando diferentes conjuntos de datos. En general, KutralNet presenta el mismo comportamiento que un modelo de aprendizaje profundo, logrando un alto rendimiento con un número reducido de parámetros y operaciones.



(a) Curva ROC para los modelos entrenados con FireNet. El modelo FireNet funciona ligeramente mejor que ResNet50, OctFiResNet y KutralNet



(b) Curva ROC para los modelos entrenados sobre FiSmo. FireNet realiza el mejor valor AUROC pero con baja precisión de prueba. KutralNet es el segundo mejor valor de AUROC que logra la mejor precisión de la prueba, seguido por OctFiResNet y ResNet50.



(c) Curva ROC para los modelos entrenados sobre FiSmoA. El rendimiento mejora con esta versión aumentada de FiSmo. FireNet presenta el mejor valor AUROC pero con baja precisión de prueba. ResNet50 obtiene el segundo mejor valor de AUROC logrando la mejor precisión de prueba, seguido por KutralNet y OctFiResNet.

Figura 5

Resultados de prueba de los modelos con el dataset FireNet-Test con 871 imágenes para la clasificación de fuego. Todos los modelos fueron entrados con diferentes conjuntos de datos. El dataset aumentado, FiSmoA, presenta mejores resultados que FiSmo, para todos los modelos.

Modelo	Test Acc	AUROC
FireNet50	64.27%	0.96
KutralNet	78.26%	0.92
OctFiResNet	75.92%	0.87
ResNet50	70.26%	0.90

Tabla IV

Valores medios de precisión de prueba e índice AUROC de cada modelo.

Nuestro modelo base KutralNet, logra un rendimiento interesante en comparación con los anteriores modelos profundos de reconocimiento de fuego. Este modelo presenta pocas capas de convolución para adquirir una representación característica del fuego en imágenes. Un modelo con pocas capas presenta, por consiguiente, un número reducido de parámetros y operaciones necesarias para esta tarea. Nuestro modelo base resultante reduce en un 85% el número de parámetros y en un 92% el número de operaciones necesarias, en comparación con el modelo OctFiResNet, para procesar una imagen de 84x84 píxeles en canales RGB.

B. Versión portátil

Con nuestra arquitectura base KutralNet, el siguiente experimento fue reducir su costo computacional. Para ello, a partir de la línea base, se modifican algunas capas de convolución, dando como resultado tres diferentes modelos para comprobar la forma más eficiente de convolución. El primer modelo, KutralNet Mobile, sustituve la estructura base superficialmente para implementar los inverted redisual block con convolución depthwise, propuesto en Ref.[3], simplificando las operaciones necesarias para el procesamiento. El segundo modelo, KutralNet Octave, sustituye la convolución regular base por la convolución octave [6] para el procesamiento de la señal de entrada. Para que funcione la convolución octave, se modifica ligeramente la conexión directa presente en la arguitectura base. Para el tercer modelo, KutralNet Mobile Octave, presenta una combinación de las convoluciones anteriormente mencionadas en conjunto con el inverted residual block.

En primer lugar, el entrenamiento se realizó con FiSmo, un conjunto de datos desequilibrados con imágenes de 2004 y 4059 para la etiqueta de fuego y no fuego, respectivamente. Para esta comparación de enfoques portátiles, se logra una precisión de

validación de 88,62%; 85,99%; 87,55% y 87,39% con KutralNet, KutralNet Mobile, KutralNet Octave y KutralNet Mobile Octave, respectivamente. Además, la precisión de prueba obtenida de los modelos entrenados con este conjunto de datos es de 74,63%; 67,28%; 72,33% y, 72,91%, respectivamente. El diagrama de barras de la Figura 6 muestra los resultados obtenidos con los otros conjuntos de datos. Para los modelos entrenados, se realizó la prueba con la imagen negra para comprobar la calidad de las características obtenidas de la imagen. Para ello, el uso del conjunto de datos FiSmoBA obtiene el menor error de clasificación en todos los modelos entrenados. Además, obtiene un ±1% de diferencia de precisión de validación en comparación con el conjunto de datos FiSmoB. Para la prueba de imagen negra, KutralNet Mobile Octave obtiene la menor clasificación errónea con un 10%; 30% y 0% para FiSmo, FiSmoB y FiSmoBA respectivamente. En general, como puede verse en la Figura 7, KutralNet Mobile Octave se comporta bien en las diferentes variaciones del dataset FiSmo. Además, el índice AUROC es incluso mejor que el modelo base con la versión equilibrada del dataset, y la versión equilibrada aumentada. En el caso de KutralNet Octave, se desempeña mejor que la versión Mobile oCtave con FiSmo y su versión aumentada balanceada. En la Tabla V están los valores medios obtenidos para la precisión de prueba y el índice AUROC para todos los conjuntos de datos de cada modelo portátil. Teniendo en cuenta el equilibrio entre el número de parámetros y las operaciones necesarias para el procesamiento de la imagen, Kutralnet Octave presenta una solución adecuada con un número menor de parámetros que KutralNet Mobile Octave e, inversamente, requiere más operaciones para su procesamiento.



(a) Precisión de validación obtenida de cada conjunto de datos por diferentes modelos portátiles. Todos los modelos presentan casi los mismos resultados.



(b) Precisión de prueba de cada modelo portátil entrenado con un dataset diferente y probado con FireNet-Test. KutralNet Mobile Octave y KutralNet Octave fueron capaces de superar los resultados del modelo base.

Figura 6

Resultados de entrenamiento obtenidos con los conjuntos de datos utilizados: FiSmo, FiSmoB, y FiSmoBA, una variante con 98 imágenes de la etiqueta no fuego reemplazadas con imágenes en negro. Los conjuntos de datos fueron divididos en 80/20 para validación en todos los modelos. El aumento de las imágenes con negro redujo la diferencia en la distancia de la precisión entre los modelos.

Modelo	Test Acc	AUROC
KutralNet	76.01%	0.86
KutralNet Mobile	71.99%	0.85
KutralNet Octave	73.90%	0.85
KutralNet Mobile Octave	79.49%	0.90

Tabla V

Valores medios de precisión de prueba e índice AUROC de cada modelo portátil.

Conclusiones

En este trabajo hemos propuesto un enfoque ligero para el reconocimiento de fuego, que consiste en 138,9K parámetros y 76,9M flops utilizado como base para construir tres versiones portátiles. KutralNet Mobile Octave, consiguió obtener un buen rendimiento requiriendo sólo de 24,6M de flops con 185,3K parámetros, siendo adecuado para dispositivos móviles.



(a) Curva ROC para los modelos entrenados sobre FiSmo. KutralNet Octave obtiene el primer lugar, seguido por KutralNet. KutralNet Mobile Octave y KutralNet Mobile se quedan por debajo del rendimiento de KutralNet.



(b) Curva ROC para los modelos entrenados sobre el dataset balanceado de FiSmo. KutralNet Mobile Octave supera los resultados de KutralNet seguido por la versión Mobile.



(c) Curva ROC para los modelos entrenados sobre FiSmoBA. Solo la versión Octave y Mobile Octave de Kutralnet superan el modelo base.

Figura 7

Resultados de entrenamiento obtenidos con los conjuntos de datos utilizados: FiSmo, FiSmoB, y FiSmoBA, una variante con 98 imágenes de la etiqueta no fuego reemplazadas con imágenes en negro. Los conjuntos de datos fueron divididos en 80/20 para validación en todos los modelos. El aumento de las imágenes con negro redujo la diferencia en la distancia de la precisión entre los modelos.

Este trabajo ha sido financiado en parte por la 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), y el Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), agencias de investigación de Brasil.

Referencias

[1] Y. LeCun, et al., "Deep learning," Nature, vol. 521, no. 7553, pp. 436–444, may 2015.

[2] Y. Deng, "Deep learning on mobile devices: A review," in Mobile Multimedia/Image Processing, Security, and Applications 2019, pp. 52 – 66.

[3] M. Sandler, et al., "MobileNetV2: Inverted residuals and linear bottlenecks," in 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, June 2018, pp. 4510–4520.

[4] X. Zhang, et al., "Shufflenet: An extremely efficient convolutional neural network for mobile devices," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 6848–6856.

[5] B. Zoph, et al., "Learning transferable architectures for scalable image recognition," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 8697–8710.

[6] Y. Chen, et al., "Drop an octave: Reducing spatial redundancy in convolutional neural networks with octave convolution," arXiv preprint arXiv:1904.05049, 2019.

[7] A. Jadon, et al., "FireNet: A specialized lightweight fire & smoke detection model for real-time iot applications," CoRR, vol. abs/1905.11922, 2019.

[8] K. He, et al., "Deep residual learning for image recognition," CoRR, vol. abs/1512.03385, 2015.

[9] B. Xu, et al., "Empirical evaluation of rectified activations in convolutional network," CoRR, vol. abs/1505.00853, 2015.

[10] M. T. Cazzolato, et al., "Fismo: A compilation of datasets from emergency situations for fire and smoke analysis," in Brazilian Symposium on Databases - SBBD, 2017, pp. 213–223.

[11] J. Sharma, et al., "Deep convolutional neural networks for fire detection in images," in Engineering Applications of Neural Networks. Cham: Springer International Publishing, 2017, pp. 183–193.

[12] A. Ayala, et al., "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.