5G AI-IoT system for bird species monitoring and bird song classification

Jaume Segura-Garcia*
Miguel Arevalillo-Herraez*
Santiago Felici-Castell*
Enrique A. Navarro-Camba*
Computer Science Dpt
Burjassot, Valencia, Spain
jaume.segura@uv.es
miguel.arevalillo@uv.es
santiago.felici@uv.es
enrique.navarro@uv.es

Sean Sturley*
Jose M. Alcaraz-Calero*
School of Computing, Engineering and Physical Sciences,
University of the West of Scotland,
Paisley, Renfrewshire, Scotland, UK
sean.sturley@uws.ac.uk
jose.alcaraz-calero@uws.ac.uk

ABSTRACT

Identification of animal species is a crucial aspect of biology and ecology, particularly in ornithology, where collaboration with other disciplines aims to develop effective methods for bird protection and environmental quality assessment. Leveraging artificial intelligence (AI) and Internet of Things (IoT) technologies, advancements in birdsong identification have been achieved. Machine learning and deep learning techniques, including Imagenet-based Convolutional Neural Networks (CNNs) like EfficientNet and MobileNet, have been employed for image feature comparison. Spectrograms of birdsongs have been analyzed, with Deep CNNs (DCNNs) proving effective in reducing model size for birdsong classification.

A 5G IoT-based system for raw audio collection has been implemented, testing various CNNs for bird identification from audio recordings. While Imagenet-based CNNs exhibit high accuracy (up to 75%), their training requires a substantial number of parameters, hindering efficiency during inference. In response, two Deep CNNs were designed to reduce parameters while maintaining accuracy, enabling integration into Small Board Computers (SBCs) or Microcontroller Units (MCUs). These DCNNs demonstrated a 6.7% improvement in overall accuracy compared to other networks, offering a lighter solution for deployment in SBCs and MCUs.

CCS CONCEPTS

• **Computing methodologies** → **Artificial intelligence**; **Classification and regression trees**; • **Networks** → *Sensor networks*; • **Hardware** → **Wireless integrated network sensors**.

*Authors contributed equally to this research.

This work is licensed under a Creative Commons Attribution International 4.0 License.

EATIS 2024, July 03–05, 2024, Praia, Cape Verde
© 2024 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-1733-8/24/07
https://doi.org/10.1145/3685243.3685244

KEYWORDS

Artificial Intelligence (AI), Internet of Things (IoT), birdsong monitoring and classification, deep learning, convolutional neural networks, wireless acoustic sensor networks

ACM Reference Format:
Jaume Segura-Garcia, Miguel Arevalillo-Herraez, Santiago Felici-Castell, Enrique A. Navarro-Camba, Sean Sturley, and Jose M. Alcaraz-Calero. 2024. 5G AI-IoT system for bird species monitoring and bird song classification. In *12th Euro American Conference on Telematics and Information Systems (EATIS 2024), July 03–05, 2024, Praia, Cape Verde.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3685243.3685244

1 INTRODUCTION

La evaluación de la calidad ambiental como cuestión principal en la construcción de un ecosistema ecológico es un deber importante de toda la sociedad. En estos ecosistemas (ya sean urbanos o naturales), las aves desempeñan un papel esencial y la formación de comunidades de aves puede utilizarse como índice paramétrico de la calidad ambiental [18].

La norma sobre paisajes sonoros, ISO 12913 [9], en su segunda parte, describe una taxonomía para la clasificación de los sonidos, que tiene en cuenta la tipología de la fuente sonora y del entorno, que también se especifica en la taxonomía de los conjuntos de datos URBANSOUND [22], pero está orientada únicamente a los sonidos urbanos [20]. El entorno natural también forma parte de una visión más general de esta taxonomía, ya que considera otros sonidos aparte de los producidos en una visión centrada en el ser humano.

En el medio natural, la exploración de la avifauna reviste gran importancia en los ámbitos de la conservación de las aves, la comprensión del ecosistema de los humedales y la evaluación de la calidad del ecosistema. Según informa la Unión Internacional para la Conservación de la Naturaleza (UICN), hay cientos de especies de aves diferentes en todo el mundo, un porcentaje considerable de ellas al borde de la vulnerabilidad y la extinción inminente [17]. Debido a las notables características de las aves, como su excepcional movilidad, su extensa área de distribución geográfica y su gran adaptabilidad a diversos entornos, su canto, que es un signo revelador de sus actividades, se utiliza con frecuencia para la detección, el seguimiento y la cuantificación de las especies. El canto de las aves presenta rápidas variaciones temporales, ofreciendo

estabilidad dentro de la misma especie y claras distinciones entre especies. El desarrollo de modelos automatizados de clasificación de aves a partir de datos sonoros derivados del canto de las aves es muy prometedor para numerosas aplicaciones en los ámbitos de la conservación de las aves, la ecología y el archivo de datos.

En este contexto, el despliegue de redes inalámbricas de sensores acústicos (WASN) resulta muy prometedor, dada su capacidad para recoger datos y realizar mediciones de forma descentralizada. Además, ofrecen adaptabilidad y fácil reconfiguración para dar cabida a mediciones adicionales e integrar nuevos parámetros.

Una WASN consta de varios nodos, cada uno con su propia unidad de procesamiento, memoria y módulos de comunicación inalámbrica, que suelen utilizar tecnología WiFi. Estos nodos pueden conectarse a micrófonos, lo que les confiere una funcionalidad versátil. Generalmente rentables, estas redes permiten la expansión flexible de los nodos para satisfacer requisitos específicos o escenarios de prueba. Los nodos se construyen a menudo en torno a ordenadores de placa única (SBC) adaptables, como el bien establecido Raspberry Pi (RPi) [19]. Los nodos conectados pueden grabar audio con frecuencias de muestreo y profundidades de bits por muestra variables.

Estas redes encuentran diversas aplicaciones, incluyendo tareas como la localización de fuentes de sonido [1], el seguimiento de fuentes de sonido [14], la identificación de fuentes de sonido [5], y la medición de características ambientales específicas [16], lo que demuestra su versatilidad en diversos dominios.

Este artículo describe el desarrollo de un sistema 5G-IoT diseñado para grabar el canto de las aves, profundizando más allá de los hallazgos de Liu et al.[12]. La ampliación de este estudio implica la recopilación de un conjunto más amplio de especies de aves, que abarca hasta 41 especies. Se realiza un análisis comparativo entre el conjunto de datos un conjunto aumentado en el que se emplean técnicas temporales como la adición de ruido, el estiramiento, el desplazamiento temporal y el desplazamiento de tono. En la fase de inferencia se introducen técnicas de postprocesamiento para mejorar la precisión de la detección. El objetivo general es aumentar el rendimiento de la aplicación BirdNet, mediante el entrenamiento de redes neuronales más ligeras que conduzcan a sistemas de monitorización 5G IoT más amplios.

Tras esta introducción, el artículo se organiza de la siguiente manera. La sección 2 muestra el trabajo relacionado. La sección 3 describe los materiales y métodos empleados, proporcionando información sobre las tecnologías y la arquitectura del sistema IoT, así como detalles sobre el conjunto de datos y las técnicas de IA (basadas en CNN) objeto de comparación. La sección 4 presenta los resultados derivados del entrenamiento de las CNNs seleccionadas para la clasificación del canto de los pájaros, junto con las predicciones realizadas utilizando el sistema AI-IoT integrado con sensores y microcontroladores de bajo coste (en concreto, el micrófono INMP441 y el ESP32-S3 [8, 6]). Por último, la sección 5 concluye el artículo.

2 ESTADO DEL ARTE

La Inteligencia Artificial (IA) unida a los sistemas de Internet de las Cosas (IoT) han encontrado diversas aplicaciones en diversos ámbitos. Estudios recientes se han dedicado a investigar diferentes enfoques para minimizar la carga computacional de algoritmos intrincados destinados a ser implementados en nodos de Internet de las Cosas (IoT) dentro de aplicaciones de redes inalámbricas de sensores acústicos (WASN).

Un ejemplo notable de la simbiosis entre IA e IoT es el presentado por López et al. en su trabajo de 2023 [13]. En este estudio, se utilizan sistemas IoT basados en Raspberry Pi para recopilar datos acústicos de salas en aulas, lo que facilita la determinación de métricas acústicas para cada espacio. Estas tecnologías integradas contribuyen al establecimiento de sistemas de monitorización rentables.

Además, se han empleado varias arquitecturas de redes neuronales convolucionales (CNN) para estimar diversos parámetros acústicos de salas. En el trabajo presentado en [7], se utiliza una CNN para la estimación ciega del volumen de la sala a partir de señales de voz monocanal reverberantes en presencia de ruido. Del mismo modo, en [23], se introduce una CNN para la estimación de la geometría de la sala y los coeficientes de reflexión basados en las respuestas al impulso de la sala (RIR), logrando una precisión de 6,5 cm para cada dimensión en la geometría de la sala y una precisión de 0,09 en los coeficientes de reflexión. En particular, este enfoque no requiere el conocimiento de las posiciones relativas de las fuentes y los receptores en la sala. Además, en [15], se demuestra la utilización del aprendizaje profundo en la clasificación de una grabación basada en la sala en la que se grabó, aprovechando el efecto de la reverberación en el habla, con una precisión máxima del 90%.

La integración de la Inteligencia Artificial (IA) y el Internet de las Cosas (IoT) trasciende los entornos urbanos. En el ámbito de la e-Salud, Khan et al. (2022) [11] proporcionan una amplia visión general de las aplicaciones del sistema AI-IoT para analizar, monitorizar y evaluar las métricas de salud, en particular durante la pandemia COVID-19. Además, en el ámbito de la Agricultura 5.0, Debauche et al. (2020) [4] aplicaron tecnologías de IoT e IA para discernir y regular los niveles de riego, identificar enfermedades de las plantas y gestionar infestaciones de plagas. Bedi et al. (2022) [2] abordan la gestión inteligente de la energía en zonas urbanas, explorando la eficacia de las técnicas de aprendizaje automático e IoT para este fin específico.

Además, un caso destacable más allá de estas investigaciones es la creación de BirdNet por parte de la Universidad de Cornell. Esta aplicación se basa en la arquitectura de redes neuronales EfficientNetB0 [10], facilitando la clasificación de aves. Accesible tanto a través de teléfonos móviles como de un sitio web específico, esta innovadora herramienta ejemplifica las diversas e impactantes aplicaciones que surgen de la integración de las tecnologías de IA e IoT en diversos ámbitos.

Los trabajos anteriores confirman que la monitorización y clasificación de sonidos de animales son problemas clásicos que tanto han atraído el interés de la comunidad investigadora, como que siguen demandando soluciones innovadoras y desafiantes para la era AI-IoT, como la que se propone en este trabajo.

3 MATERIALES Y MÉTODOS

En esta sección, proporcionaremos información sobre el desarrollo de la arquitectura del sistema IoT, que recopilará información de audio sobre el terreno. Este sistema debe proporcionar una gestión

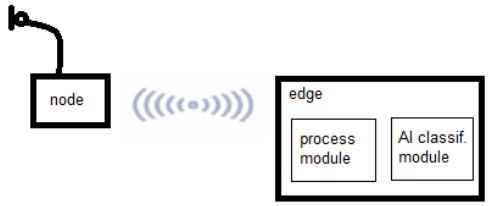

Figure 1: Esquema del sistema AI-IoT.

eficiente de la energía con el fin de permitir un período de recogida de datos. En este caso, utilizaremos un nodo basado en la arquitectura de la unidad microcontroladora (MCU) ESP32, ya que se ha demostrado [3] que es una opción rentable y potente (en el caso de la MCU ESP32-S3) para la recogida y preprocesado de audio [21].

3.1 Arquitectura del sistema IoT

El sistema IoT se basa en un nodo para la recogida de información de audio y un EDGE para el procesamiento del audio y para la clasificación de las aves con el módulo de IA. La figura 1 muestra un esquema del sistema AI-IoT, donde el EDGE se basa en una Raspberry Pi 4B+ con 8 GB de RAM. En esta figura, se muestran el módulo de procesamiento y el módulo de clasificación de IA ubicados en el EDGE.

La arquitectura del nodo IoT para la grabación del canto de los pájaros se basa en un micrófono MEMS (INMP441) y una unidad microcontroladora (MCU, en este caso Lilygo T-SIM7080G basada en ESP32-S3 WROOM-1-N16R8). La MCU utilizada lleva integrado un SoC de la serie ESP32-S3, microprocesador Xtensa dual-core LX7 de 32 bits (con hasta 240 MHz). Esta MCU tiene hasta 8MB de PSRAM y flash de hasta 16MB. Esta MCU también dispone de comunicaciones inalámbricas con interfaz WiFi (802.11 b/g/n), BLE (con antena embebida), LTE-Cat-M y NB-IoT (con antena externa y conexión a tarjeta SIM) y diferentes periféricos (I2C, SPI, UART, SDIO, I2S, CAN)[1] [6]. También permite la conexión GNSS/GPS y la posibilidad de añadir almacenamiento externo con una tarjeta microSD. La figura 2 muestra una fotografía del nodo con el micrófono y la MCU. Esta MCU ha sido programada con MQTT, para publicar audio, que será recogido en el EDGE (Raspberry Pi 4B+[19]) para recoger audio y realizar su posterior procesado.

3.2 Dataset y técnicas de aumento de datos

El dataset se ha obtenido de la página web de la Fundación Xeno-canto y del Centro de Biodiversidad Naturalis (https://xeno-canto.org). Esta fundación mantiene un proyecto de recopilación de sonidos de aves de todo el mundo. De este sitio web, hemos seleccionado 41 especies de aves que podrían ser detectadas e identificadas dentro de nuestro sistema. Estos audios recogidos para nuestro conjunto de datos se han grabado con una frecuencia de muestreo de 44100 y 16 bits, y se han seleccionado como especies de aves comunes en Europa, que en muchos casos forman parte del

[1] https://www.cnx-software.com/2023/03/04/esp32-s3-sim7080g-board-wifi-bluetooth-nb-iot-cat-m-gps/ (Consultado el 23/10/2023)

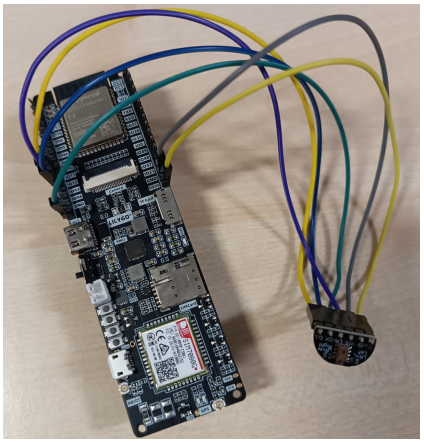

Figure 2: Fotografía de la MCU ESP32-S3 conectada al micrófono INMP441 (nodo).

entorno de nuestras ciudades. Este conjunto de datos contiene más de 1300 horas de grabaciones de audio en formato MP3 (64 GB).

Para la etapa de preprocesado, se han utilizado trozos de audio de 5s, procesados como espectrograma de coeficientes cepstrales de frecuencia Mel (MFCC), estableciendo sus parámetros como el número de FFT, el tamaño de ventana, y el tamaño de salto a 1024, el número de coeficientes Mel a 128, el tipo de ventana a Hanning.Cada espectrograma se ha guardado con un formato 'Portable Network Graphics' (denominado como imagen 600x600 png), que posteriormente se utilizará para entrenar, validar y probar las diferentes redes neuronales.

En la tabla 1, la segunda columna muestra el número de archivos de audio obtenidos del sitio web mencionado para cada especie de ave. La tercera columna muestra el número de imágenes de espectrogramas (basados en MFCC) obtenidos a partir de los archivos de audio para cada especie (aquí, cada espectrograma toma una ventana de audio de 5s).

En la cuarta y quinta columnas de la Tabla 1 hemos aumentado el conjunto de datos con técnicas de aumento temporal de audio (como adición de ruido blanco, estiramiento temporal, desplazamiento de tono, desplazamiento temporal y combinaciones de ellas). Se han implementado utilizando las bibliotecas *librosa* y *audiomentation* en Python 3.8. Llegados a este punto, es necesario aclarar que el algoritmo utilizado para el aumento de audio se basa en el número de archivos de audio, es decir si el número de archivos de audio es inferior a 200, el aumento se realiza mediante "adición de ruido", "time streching", "pitch shifting", "time shifting", "combinación de adición de ruido + time streching", "combinación de adición de ruido + pitch shifting", "combinación de adición de ruido + time shifting" y "combinación de adición de ruido + streching + time shifting"; si el número de archivos de audio está entre 200 y 400 archivos, el aumento se realiza con "añadir ruido", "time streching", "pitch shifting", "time shifting", "combinado añadir ruido + time streching", "combinado añadir ruido + pitch shifting", "combinado añadir ruido + time shifting"; si el número de archivos de audio está entre 400 y 600 archivos, el aumento se realiza con "añadir ruido", "time streching", "pitch shifting", "time shifting"; si el número de archivos de audio

está entre 600 y 800 archivos, el aumento se realiza con "añadir ruido", "pitch shifting", "time shifting"; si el número de archivos de audio está entre 800 y 1000, el aumento se realiza mediante "adición de ruido", "pitch shifting"; si el número de archivos de audio está entre 1000 y 1500, el aumento se realiza mediante "adición de ruido" y no se aplica ninguna técnica si el número de archivos de audio es superior a 1500. Con este aumento, el conjunto de datos ha aumentado considerablemente y tiene la oportunidad de equilibrarse. La figura 3 muestra algunos ejemplos de espectrogramas de canto de aves con las correspondientes técnicas de realce aplicadas, prestando atención a los criterios explicados anteriormente.

3.3 Diferentes arquitecturas de redes neuronales

La plataforma hardware utilizada para el entrenamiento y experimentación de las diferentes arquitecturas CNN es un ordenador de sobremesa con 32 GB de memoria, Intel Core i9 7900X con CPU de 10 núcleos y 20 hilos, 3,30 GHz de frecuencia, y GPU 3060 de 12 GB. El sistema operativo es Windows 10 Professional de 64 bits. También utilizamos Anaconda3, Tensorflow 2.10 y Python 3.8 como plataforma de aprendizaje profundo.

Para la implementación del subsistema de IA, hemos establecido una comparación de diferentes opciones para el desarrollo del módulo de clasificación de IA para las aves. Estas opciones aquí consideradas se basan en la necesidad de movilidad de los dispositivos utilizados (lo que implica un bajo número de parámetros para la inferencia), que pueden ser dispositivos de placa única (SBC), como Raspberry Pi 4B+ [19] o unidades microcontroladoras (MCU) con memoria suficiente para contener todo el fichero de pesos de la CNN, como ESP32-S3 [6].

Para la adopción de este paradigma, hemos seleccionado, por un lado, redes neuronales convolucionales (CNN) ligeras basadas en arquitecturas entrenadas con ImageNet, como EfficientNet-B0 y -B4, y EfficientNetV2-Small y V2-B0, o MobileNet-v2 y -v3 Large y Small, y por otro lado, una arquitectura CNN ad hoc utilizada para clasificar estos espectrogramas. Esta selección se basó en la necesidad de tener un archivo ligero para almacenar los pesos de la red para la etapa de inferencia en un SBC (como Raspberry Pi 4B+) o incluso en una MCU (como ESP32-S3 con 16MB flash y 8MB PSRAM, por lo que necesitamos conseguir alguna red más ligera que 4MB para que este dispositivo pueda manejarse con ella). Tabla 2 muestra una comparación de estas redes con el número de parámetros y el peso del archivo resultante (con capas aplanadas y densas en la salida). Nuestro propósito aquí es encontrar una Red Neuronal Convolucional Profunda (DCNN) adecuada para ser utilizada en SBC o en MCU. Según esta tabla, las CNNs basadas en los pesos de Imagenet serán adecuadas para SBC, como Raspberry Pi 4B+ con 4GB, incluso la llamada DCNN 1, aunque encajaría correctamente en una MCU, como ESP32-S3 con 8MB PSRAM, pero el Sistema Operativo y la DCNN 1 serían demasiado para esta MCU. La mejor opción para la MCU es el DCNN 2, que cabe bien y tiene suficiente memoria RAM para el procesamiento posterior.

4 RESULTADOS

En esta sección, explicamos el proceso de entrenamiento y validación, con el conjunto de datos normal y el conjunto de datos aumentado. A partir de este proceso, vimos que el conjunto de datos aumentado daba mejores resultados en todos los casos si dejamos todas las capas entrenables disponibles para el entrenamiento. Como veremos, con este aprendizaje de transferencia (utilizando MobileNet V2, V3S y V3L y EfficientNet B0, B4, V2S y V2B0), somos capaces de alcanzar una precisión suficiente para distinguir el canto de una especie de ave concreta, pero no podemos alcanzar la precisión del estado del arte de Liu2022, que obtuvo una precisión del 89,6% con 30 especies de aves, con una CNN multiescala, considerando varias opciones para el kernel en la capa Conv2D.

4.1 Evaluación de diferentes predicciones

Tras el entrenamiento y la evaluación de las distintas redes, como el audio suele ser más largo que los 5s que utilizamos para la imagen del espectrograma, hemos diseñado un algoritmo para postprocesar distintos espectrogramas en la fase de prueba reuniendo los archivos con el mismo nombre de prefijo, con el fin de juntar la secuencia temporal de los espectrogramas. Así, juntando las predicciones en una matriz, podemos votar por la clase preferida y evaluar la correcta. De este modo, también podemos calcular la precisión de este proceso. La figura 6 muestra un esquema de este algoritmo de votación.

En la Figura 7, comparamos la precisión global media obtenida en el proceso de entrenamiento para todo el conjunto de clases de aves (41) y el uso de este algoritmo. Tras aplicar este procedimiento, observamos que la acumulación de predicciones consecutivas permite aumentar la precisión en la evaluación de la clase de un ave. Por lo tanto, en el dispositivo IoT, grabaremos durante al menos 15s para permitir este post-procesamiento. Esta propuesta puede ser completamente funcional para cualquier escenario.

Para obtener una versión reducida de cada modelo, vamos a utilizar Tensorflow-Lite (tflite) para la integración en SBC (como Raspberry Pi) o en MCU (como ESP32-S3). En nuestro caso, la Tabla 7 muestra los pesos resultantes de los ficheros tras el proceso de conversión, sin aplicar ninguna cuantización para evitar cualquier pérdida en la precisión.

En la tabla 7 observamos que el único modelo disponible para una MCU, como ESP32-S3, es Deep CNN 2, con 4 capas convolucionales, que se describe en la figura 5. En este caso, hay que tener en cuenta que el resto de la memoria PSRAM se necesita para otros procesos del Sistema Operativo (aquí hay que recordar que el ESP32-S3 puede tener 8 MB de PSRAM). Así mismo, el hecho de usar truncamiento en estas redes (DCNN) nos permite que no haya pérdida significativa de precisión (menor del 3%). El resto de ConvNets se podrían utilizar en SBC, como Raspberry Pi 4B+ con 4 u 8GB, ya que no caben correctamente en la RAM de cualquier MCU de bajo consumo.

5 CONCLUSIONES

En este trabajo, hemos desarrollado un sistema de monitorización basado en tecnologías LTE-M/NB-IoT basadas en 5G para Comunicaciones Masivas de Tipo Máquina (mMTC), combinado con un módulo de Inteligencia Artificial que permite la clasificación de hasta 41 especies de aves. Para el desarrollo de este módulo, hemos seleccionado diferentes ConvNets, basadas en EfficientNet y MobileNet con pesos de ImageNet, y hemos comparado su precisión en entrenamiento, validación y test.

Table 1: Tabla con las distintas especies de aves consideradas, con el número de archivos de audio recogidos y el número de imágenes de espectrogramas (cada una tiene información de 5s de audio). Consideramos el conjunto de audio normal y el conjunto de audio aumentado.

Especie	Nr audio	Nr muestras espectr.	Nr audio augm.	Nr muestras espectr. augm.
Alauda arvensis	1447	22941	2897	45882
Alectoris rufa	103	1129	928	10161
Apus apus	48	689	433	6201
Athene noctua	286	5444	2033	38108
Carduelis carduelis	1079	16489	2161	33340
Chloris chloris	1065	14585	2133	29178
Coccothraustes coccothraustes	186	2098	1675	18882
Columba palumbus	498	6568	2492	32840
Corvus cornix	188	2206	1693	19862
Corvus corone	63	662	568	5958
Corvus frugilegus	42	685	379	6165
Delichon urbicum	89	2043	802	18387
Dendrocopos major	50	542	451	4878
Emberiza citrinella	1886	44867	1890	44867
Erithacus rubecula	2660	72123	2666	72123
Fringilla coelebs	3254	61676	3260	61676
Garrulus glandarius	213	3629	1492	25403
Hirundo rustica	753	13869	3014	55479
Luscinia luscinia	607	17789	2427	71471
Motacilla alba	217	2185	1520	15295
Otus scops	353	4574	2472	32018
Parus major	2858	41953	2864	41953
Passer domesticus	665	10419	2662	41676
Passer montanus	305	7694	2136	53858
Perdix perdix	93	1162	838	10458
Phoenicurus ochruros	679	11193	2718	44772
Phoenicurus phoenicurus	847	17890	2543	53670
Phylloscopus collybita	2475	48218	2481	48218
Phylloscopus trochilus	3394	111706	3401	111706
Pica pica	84	1137	757	10233
Riparia riparia	36	918	325	8262
Sitta europaea	534	6081	2672	30405
Streptopelia decaocto	519	6253	2597	31265
Sturnus unicolor	176	2382	1585	21438
Sturnus vulgaris	696	18964	2786	75856
Troglodytes troglodytes	2198	40168	2203	40168
Turdus merula	3195	127186	3202	127186
Turdus philomelos	3095	140193	3102	140193
Turdus pilaris	190	2739	1711	24659
Turdus torquatus	165	4366	1486	39294
Tyto alba	73	1515	658	13635
TOTAL	37392	899108	80366	1618681

En este estudio, también hemos propuesto el aumento del conjunto de datos original con la aplicación de diferentes técnicas de procesamiento de señales en el dominio temporal a los audios originales. Para el conjunto de datos aumentado, hemos obtenido mejores resultados de precisión y f1-score que para el conjunto de datos original. Además observamos que para nuestro conjunto de datos aumentado, la mejor opción de la familia EfficientNet es EfficientNet v2 Small (71,4% de precisión), pero es más pesada que la siguiente red, que es EfficientNet v2 B0 (69,83% de precisión). Para la familia MobileNet, la mejor opción es MobileNet v2, con un 68,58% de precisión. Pero, la mejor opción es en todos los casos DCNN de 4 capas, que dependiendo de la estructura del número de filtros puede tener un peso mayor. En este caso, la opción más rentable para su uso en un SBC es la DCNN 1 (72,95% de precisión), con una estructura de 128-64-32-16 filtros (mostrada en la Figura 4), pero la mejor opción para una MCU es la DCNN 2 (63,49% de

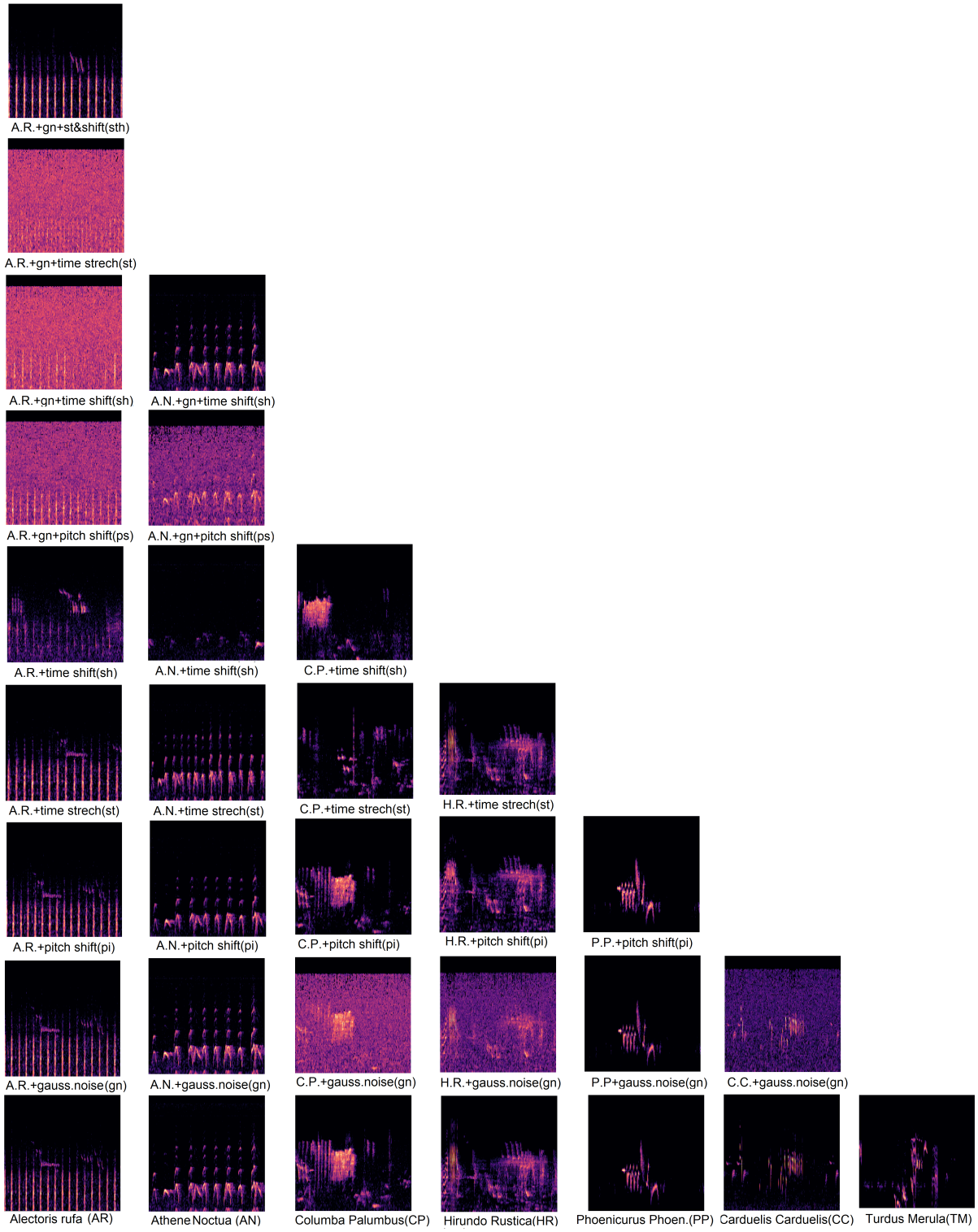

Figure 3: Ejemplos de espectrogramas de especies de aves con técnicas de aumento de audio según los criterios establecidos.

Table 2: Comparación de ConvNets según el número de parámetros

ConvNet	Nr de parametros
EfficientNet v2S	20,383,881
EfficientNet B4	18,144,112
EfficientNet v2 B0	6,595,705
MobileNet v3L	4,925,033
MobileNet v2	4,829,545
EfficientNet B0	4,387,788
MobileNet v3S	2,096,345
Deep CNN 1 (Figure 4)	1,301,657
Deep CNN 2 (Figure 5)	331,745

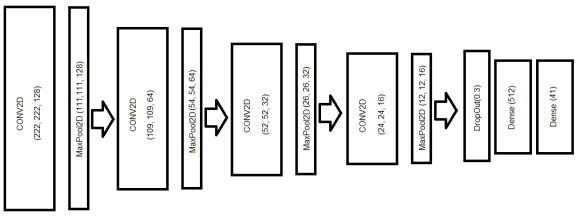

Figure 4: Estructura de la Deep CNN propuesta con 4 capas convolucionales con MaxPool (con 128, 64, 32 y 16 filtros, respectivamente).

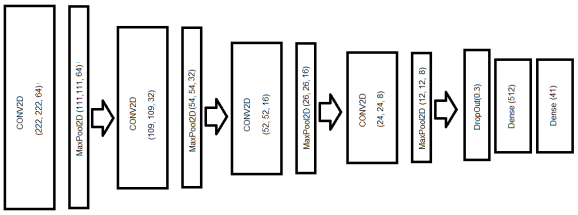

Figure 5: Estructura de la Deep CNN propuesta con 4 capas convolucionales con MaxPool (con 64, 32, 16 y 8 filtros, respectivamente).

Table 3: Definición de capas de red EfficientNet B0

Capa (tipo)	Forma de Salida	Param #
EfficientNet B0 (Func.)	(None, 7, 7, 1792)	4049571
GlobalAveragePooling2D	(None, 1792)	0
Dense	(None, 256)	327680
Dropout (0.3)	(None, 256)	0
Dense	(None, 41)	10537
Total params:		4,387,788 (16.74 MB)

precisión), con una estructura de 64-32-16-8 filtros (mostrada en la Figura 5). Las Tablas 3, 4, 5, y 6 muestran la estructura de capas de salida, mediante aprendizaje transferido (Transfer Learning) de varias familias de CNNs ampliamente usadas en entornos IoT por

Table 4: Definición de capas de red EfficientNetV2-B0

Capa (tipo)	Forma de Salida	Param #
EfficientNetV2-B0 (Func.)	(None, 7, 7, 1280)	25919312
GlobalAveragePooling2D	(None, 1280)	0
Dense	(None, 256)	327680
Dropout (0.3)	(None, 256)	0
Dense	(None, 41)	10537
Total params:		6,257,529 (18.78 MB)

Table 5: Definición de las capas de red de MobileNetV2

Capa (tipo)	Forma de Salida	Param #
MobileNetV2 1.0.224 (Func.)	(None, 7, 7, 1280)	2257984
Flatten (Flatten)	(None, 62720)	0
Dropout (0.3)	(None, 62720)	0
Dense	(None, 41)	2571561
Total params:		4,829,545 (18.42 MB)

Table 6: Definición de capas de red MobileNetV3 (small)

Capa (tipo)	Forma de Salida	Param #
MobileNetV3 - small (Func.)	(None, 7, 7, 576)	939120
Flatten (Flatten)	(None, 28224)	0
Dropout (0.3)	(None, 28224)	0
Dense	(None, 41)	1157225
Total params:		2,096,345 (8.00 MB)

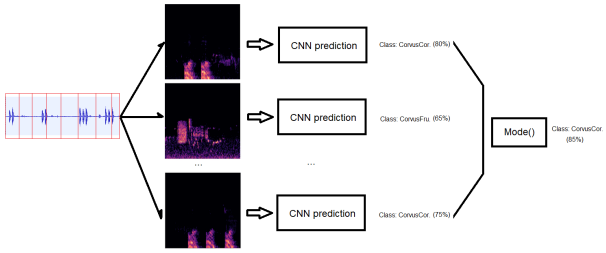

Figure 6: Esquema del algoritmo de votación para el posprocesamiento de predicciones con el fin de aumentar la precisión global de las distintas predicciones.

su bajo peso. Además, proponemos un algoritmo de votación tras el paso de predicción que permite una mejora de alrededor del 22% en la precisión global.

Para futuros desarrollos, trabajaremos con ConvNets para trabajar directamente con trozos de audio. Probablemente, conseguirán redes más ligeras que permitirán una aplicación directa a MCUs con suficiente RAM.

ACKNOWLEDGMENTS

Los autores agradecen a la Agencia Estatal de Investigación (AEI) y al Fondo Europeo de Desarrollo Regional (FEDER) la financiación

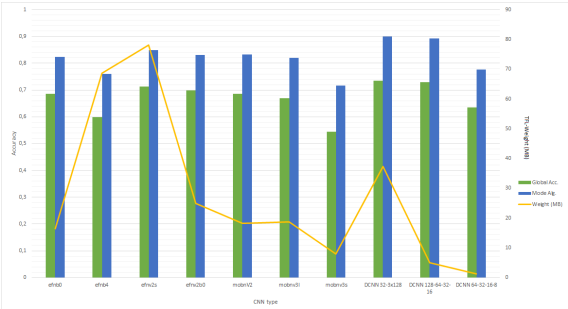

Figure 7: Comparación de la precisión global en la prueba para las diferentes redes seleccionadas y diseñadas y el peso del archivo TensorFlow-Lite.

Table 7: Comparación de las ConvNets convertidas a TfLite a partir de la tabla 2.

Red Neuronal	Peso TF-Lite
EfficientNet v2 - Small	78.1 MB
EfficientNet B4	68.6 MB
Deep CNN 0	37.2 MB
EfficientNet v2 B0	24.9 MB
MobileNet V3 - Large	18.8 MB
MobileNet V2	18.2 MB
EfficientNet B0	16.5 MB
MobileNet V3 - Small	8.0 MB
Deep CNN 1	5.0 MB
Deep CNN 2	1.3 MB

parcial de esta investigación dentro de los proyectos con referencias de subvención PID2021-126823OB-I00 financiado por MCIN/AEI/10.13039/501100011033 y por la "Unión Europea NextGenerationEU/PRTR", TED2021-131040B-C33 financiado en el programa "Proyectos Estratégicos para la Transición Ecológica y Digital 2021" por MCIN/AEI/10.13039/501100011033 y la Unión Europea NextGenerationEU/PRTR. También a la Generalitat Valenciana, por la financiación del proyecto CIAICO/2021/057 y CIAICO/2022/179. Finalmente, al Ministerio de Universidades por la financiación de la beca de estancia de investigación PRX22/000503.

REFERENCES

[1] A. Alexandridis and A. Mouchtaris. 2015. Multiple sound location estimation and counting in a Wireless Acoustic Sensor Network.. In *Proc of 2015 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics. October 18-21, 2015, New Paltz, NY.*

[2] Pradeep Bedi, S. B. Goyal, Anand Singh Rajawat, Rabindra Nath Shaw, and Ankush Ghosh. 2022. Application of AI/IoT for Smart Renewable Energy Management in Smart Cities. In *AI and IoT for Smart City Applications.* Springer Nature Singapore, Singapore, 115–138. https://doi.org/10.1007/978-981-16-7498-3_8

[3] Lukas M. Broell, Christian Hanshans, and Dominik Kimmerle. 2023. IoT on an ESP32: Optimization Methods Regarding Battery Life and Write Speed to an SD-Card. In *Edge Computing*, Sam Goundar (Ed.). IntechOpen, Rijeka, Chapter 7.

[4] Olivier Debauche, Saïd Mahmoudi, Meryem Elmoulat, Sidi Ahmed Mahmoudi, Pierre Manneback, and Frédéric Lebeau. 2020. Edge AI-IoT Pivot Irrigation, Plant Diseases, and Pests Identification. *Procedia Computer Science* 177 (2020), 40–48. https://doi.org/10.1016/j.procs.2020.10.009 The 11th International Conference on Emerging Ubiquitous Systems and Pervasive Networks (EUSPN 2020) / The 10th International Conference on Current and Future Trends of Information and Communication Technologies in Healthcare (ICTH 2020) / Affiliated Workshops.

[5] M. F. Duarte and Y. Hen Hu. 2004. Vehicle classification in distributed sensor networks. *J. Parallel Distrib.Comput.* 64 (2004), 826838. https://doi.org/10.1016/j.jpdc.2004.03.020

[6] espressif.com. 2023. ESP32-S3 datasheet. https://www.espressif.com/sites/default/files/documentation/esp32-s3_datasheet_en.pdf (Accessed: 10/11/2023).

[7] A. F. Genovese, H. Gamper, V. Pulkki, N. Raghuvanshi, and I. J. Tashev. 2019. Blind Room Volume Estimation from Single-channel Noisy Speech. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* 231–235. https://doi.org/10.1109/ICASSP.2019.8682951

[8] invensense.tdk.com. 2015. INMP441 datasheet. https://invensense.tdk.com/wp-content/uploads/2015/02/INMP441.pdf (Accessed: 11/11/2023).

[9] ISO. 2018. *Acoustics - Soundscape. Part 2, Data collection and reporting requirements (ISO/TS 12913-2:2018).* Technical Report. I.S.O. https://www.iso.org/standard/75267.html

[10] S. Kahl, C. M. Wood, M. Eibl, and H. Klinck. 2021. BirdNET: A deep learning solution for avian diversity monitoring. *Ecological Informatics* 61 (2021), 101236.

[11] Junaid Iqbal Khan, Jebran Khan, Furqan Ali, Farman Ullah, Jamshid Bacha, and Sungchang Lee. 2022. Artificial Intelligence and Internet of Things (AI-IoT) Technologies in Response to COVID-19 Pandemic: A Systematic Review. *IEEE Access* 10 (2022), 62613–62660. https://doi.org/10.1109/ACCESS.2022.3181605

[12] Jiang Liu, Yan Zhang, Danjv Lv, Jing Lu, Shanshan Xie, Jiali Zi, Yue Yin, and Haifeng Xu. 2022. Birdsong classification based on ensemble multi-scale convolutional neural network. *Scientific Reports (Nature)* 12 (2022), 8636. https://doi.org/10.1038/s41598-022-12121-8

[13] Jesus Lopez-Ballester, Santiago Felici-Castell, Jaume Segura-Garcia, and Maximo Cobos. 2023. AI-IoT Platform for Blind Estimation of Room Acoustic Parameters Based on Deep Neural Networks. *IEEE Internet of Things Journal* 10, 1 (2023), 855–866. https://doi.org/10.1109/JIOT.2022.3203570

[14] B. Malhotra, I. Nikolaidis, and J. Harms. 2008. Distributed classification of acoustic targets in wireless audio-sensor networks. *Computer Networks* (2008). https://doi.org/10.1016/j.comnet.2008.05.008

[15] Constantinos Papayiannis, Christine Evers, and Patrick A. Naylor. 2020. End-to-End Classification of Reverberant Rooms Using DNNs. *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*

28 (jan 2020), 3010–3017. https://doi.org/10.1109/TASLP.2020.3033628

[16] A. Pastor-Aparicio, J. Segura-Garcia, J. Lopez-Ballester, S. Felici-Castell, M. Garcia-Pineda, and Pérez-Solano J. J. 2020. Psychoacoustic Annoyance Implementation With Wireless Acoustic Sensor Networks for Monitoring in Smart Cities. *IEEE Internet of Things Journal* 7, 1 (2020), 128–136. https://doi.org/10.1109/JIOT.2019.2946971

[17] Nirosha Priyadarshani, Stephen Marsland, and Isabel Castro. 2018. Automated birdsong recognition in complex acoustic environments: a review. *Journal of Avian Biology* 49, 5 (2018), jav–01447. https://doi.org/10.1111/jav.01447

[18] J. Qi, S. Gage, W. Joo, B. Napoletano, and S. Biswas. 2008. Soundscape characteristics of an environment: A new ecological indicator of ecosystem health. In *Wetland and Water Resource Modeling and Assessment*. CRC Press, 201—-211.

[19] RaspberryPi.Com. 2019. Raspberry Pi 4B+. https://datasheets.raspberrypi.com/rpi4/raspberry-pi-4-datasheet.pdf (Accessed: 02/10/2023).

[20] Justin Salamon, Christopher Jacoby, and Juan Pablo Bello. 2014. A Dataset and Taxonomy for Urban Sound Research. In *Proceedings of the 22Nd ACM International Conference on Multimedia* (Orlando, Florida, USA) *(MM '14)*. ACM, New York, NY, USA, 1041–1044. https://doi.org/10.1145/2647868.2655045

[21] Jaume Segura-Garcia, Jose M. Alcaraz-Calero, Adolfo Pastor-Aparicio, Ricardo Marco-Alaez, Santiago Felici-Castell, and Qi Wang. 2021. 5G IoT System for Real-Time Psycho-Acoustic Soundscape Monitoring in Smart Cities With Dynamic Computational Offloading to the Edge. *IEEE Internet of Things Journal* 8, 15 (2021), 12467–12475. https://doi.org/10.1109/JIOT.2021.3063520

[22] UrbanSoundDataset.weebly.com. 2014. URBANSOUND8K dataset. https://urbansounddataset.weebly.com/urbansound8k.html (Accessed: 10/10/2023).

[23] Wangyang Yu and W. Bastiaan Kleijn. 2021. Room acoustical parameter estimation from Room Impulse responses using Deep Neural Networks. *IEEE/ACM Trans. Audio, Speech and Lang. Proc.* 29 (jan 2021), 436–447. https://doi.org/10.1109/TASLP.2020.3043115

Received 20 February 2024; revised 12 March 2024; accepted 5 June 2024