Sistema de detección de cotorras basado en sonido

Informe de pasantía de iniciación a la investigación - PEDECIBA Informática

Ernesto Martin Rován

Resumen

En el presente trabajo se realiza un relevamiento de algunas de las principales técnicas de procesamiento de audio para la clasificación de eventos acústicos, con el fin de estudiar la viabilidad de un sistema de detección de aves (en particular la cotorra argentina) para la protección de cultivos. El objetivo es confluir en una posible vía de desarrollo para implementar, evaluando las distintas ventajas y dificultades.

1. Introducción: Myiopsitta Monachus

La cotorra argentina (*Myiopsitta Monachus*), o perico monje (*Monk parakeet*), es una especie de ave psitaciforme de la familia *Psittacidae* originaria de América del Sur, con fuerte presencia en Uruguay, Argentina, Brasil, Paraguay y Bolivia [1]. Mide, en promedio, aproximadamente 30 cm de largo y pesa 140 g; larga cola y plumaje en tonos verdes brillantes, con alas verdes azuladas y frente, garganta, vientre y pecho en gris claro (razón por la cual ostena el nombre de "perico monje"). Se alimenta principalmente de semillas de plantas tanto silvestres como cultivadas, como el cardo, el sorgo, el maíz y el arroz. También **consume frutos** y flores, así como insectos adultos y sus larvas [2].

Figura 1: Cotorra argentina posada en eucalipto, extraido de [1]

Los pericos en general son aves considerablemente sociables, que forman sociedades complejas estructuradas en niveles, y aprenden llamados para mediar estas interacciones [7]. Las denominadas cotorras en particular, presentan un alto grado de relacionamiento, conformando parejas monogámicas estables como unidad estructural fundamental, y a su vez bandadas que en estado silvestre varían desde individuos solitarios hasta grandes grupos compuestos por colonias separadas. Su comportamiento se describe como *fission-fusion social dynamics*, en donde el tamaño y composición de la comunidad varía con el tiempo y los individuos cambian de ambiente [3], razón por la cual existen extensas poblaciones invasores en regiones del hemisferio norte, particularmente en algunos estados de EEUU y Europa. Las cotorras forman nidos que en estado nativo tienden a correlacionarse con la extensión y abundancia de arboles eucalipto, su árbol de preferencia para anidar [7].

Los patrones de fusión o separación varían según las actividades de la bandada, de hecho los tamaños de las mismas difieren significativamente dependiendo de si la bandada está posada, volando o buscando alimento [3]. Un grupo de investigadores de la American Ornithologists' Union, estudiando la estructura social de poblaciones

silvestres durante el invierno austral de 2007 en el norte de la provincia de Entre Ríos (Argentina), encontró que el tamaño promedio de las bandadas que buscaban alimento era el más grande, y esas bandadas probablemente representaban la fusión de varias bandadas más pequeñas que volaban. Sin embargo, no encontraron evidencia de que los individuos en estado salvaje compartieran información sobre la búsqueda de alimento mediante el reclutamiento vocal activo de otros directamente hacia los recursos de alimentación. Aunque la mayoría de las bandadas que volaban emitían llamados durante el vuelo, en general, provocaron tasas de respuesta muy bajas y nunca recibieron respuesta por parte de grupos que estaban buscando alimento durante sus observaciones. Ocasionalmente, las bandadas nativas que estaban en la búsqueda de alimento sí emitían llamadas de contacto, pero en general fueron menos vocales que las bandadas que volaban, y no se encontró evidencia de que la presencia de un recurso alimenticio novedoso aumentara los comportamientos de intercambio de información [3].

La comunicación se realiza mediante sonidos 'garrulos, con ásperos y **frecuentes** reclamos' [1]. Muchas investigaciones [4, 5, 6, 7] muestran fuerte evidencia de un reconocimiento individual de las cotorras entre sí mediante sus llamados, lo que indica una alta complejidad acústica en las señales que emiten. Las mismas apuntan a una *Individual vocal signature* o *Voice Print* en cada cotorra, indentificando hasta once distintos tipos de llamados [4, 8] según las circunstancias, por ejemplo: *contact call, alarm call, threth call* y distintos derivados de las posibles interacciones.

El período de mayor avistamiento de estas aves es en el invierno y otoño, pues su período de reproducción ocurre entre mediados de noviembre y principios de marzo, aunque tienen una gran presencia durante todo el año, y la diferencia de actividad no es sustancial, como muestra [11]. En la investigación citada, se releva y modelan datos de avistamiento de cotorras en Uruguay para rastrear su presencia, evolución en el tiempo y mapearla espacialmente. El mapa de calor del avistamiento de cotorras (con datos de la aplicación eBird, desarrollada por Cornell University) que se observa en la figura 2, muestra un fuerte incremento de la concentración de avistamiento de cotorras en el área metropolitana, la costa este del país y las zonas adyacentes [11].

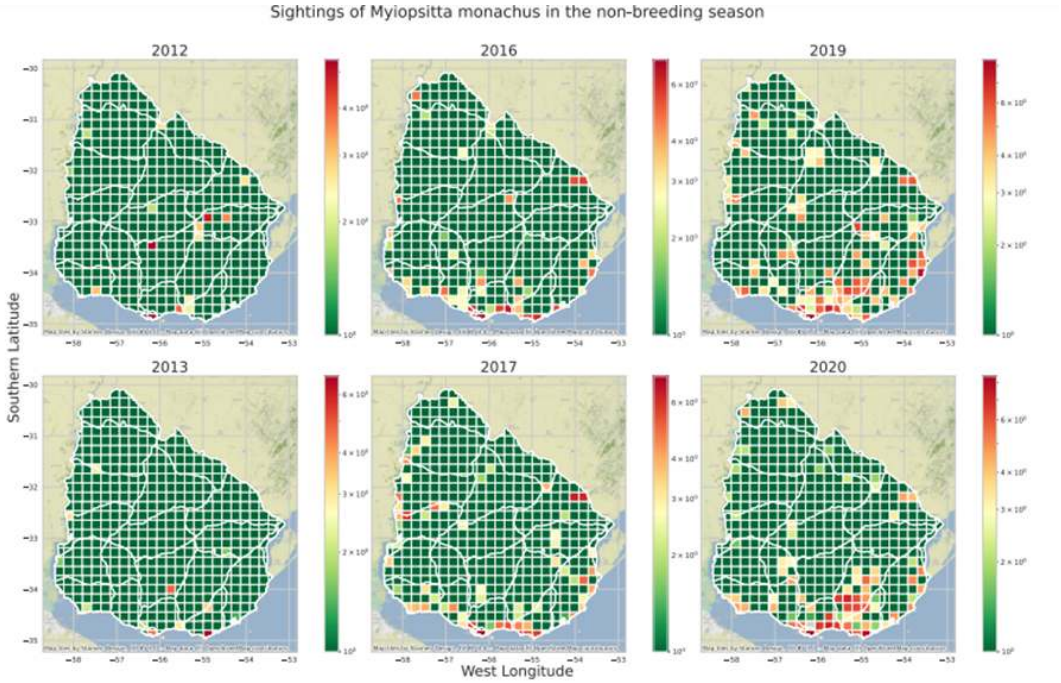


Figura 2: Avistamiento de cotorras, imagen extraida de [11]

2. Un inconveniente para el agro

Dada su adaptabilidad al medio, su gran interacción social y su amplia dieta, desde hace décadas la cotorra argentina resulta un problema para los cultivos, en particular los frutales (aunque no solo se limita a estos), impactando en la producción agrícola. Resulta interesante mencionar que en el transcurso de esta investigación se encontró un informe que data de 1973, donde un investigador de la University of Nebraska - Lincoln [9], realiza un relevamiento del problema en zonas cercanas a Montevideo, describiendo el daño que implica para los

cultivos y posibles métodos de control (los cuales parecieran no haber sido definitivos). De hecho, fue declarada plaga en 1947 y en 1981 la FAO estimó una pérdida por aves para todos los cultivos de U$ 6 millones anuales, que si bien es muy difícil de contabilizar, se calcula aún mayor en la actualidad [10].

A raíz de esto, y de su gran adaptación incluso hacia los medios tradicionales para disuadirlas (con un historial de décadas en pruebas de distintos métodos), siendo capaces de aprender la inocuidad de las potenciales amenazas presentadas, en este proyecto se plantea un sistema de disuasión de aves menos previsible basado en el vuelo de drones. El actual informe tiene como fin documentar una posible línea de técnicas para implementar la fase previa de detección y localización de las cotorras, a fin de que el sistema pueda enviar los drones a las zonas de interés de forma efectiva, haciendo viable la solución planteada.

3. Sistema de detección basado en sonido

El primer punto a considerar es que el sistema de detección debe controlar un area de considerable tamaño, que puede variar entre algunas decenas de metros cuadrados y (posiblemente) algunas hectáreas, por lo que es necesaria una red de sensores para detectar los eventos. En cuanto a la naturaleza de la señal a medir, un sistema basado en imágenes no parece viable dado el costo de equipo, de procesamiento computacional, y el rango de visión de cada cámara; ídem para otros sistemas análogos basados en sensores o cámaras térmicos. La mejor opción parece ser un sistema de sensores de sonido, con micrófonos omnidireccionales.

3.1. Un problema de clasificación

Para simplificar el problema, y comenzar por evaluar la viabilidad de detectar cotorras mediante audio, se considera el audio proveniente de una misma fuente. A partir de ahí, el problema es esencialmente de clasificación binaria, donde el sensado continuo consiste en procesar muestras de audio para determinar la presencia (*True / 1*) o no (*False / 0*) de cotorras.

Enfoque clásico

Como la señal de audio analógica se genera gracias un transductor que transforma la vibraciones mecánicas del sonido en una señal eléctrica en el tiempo (con un rango de amplitudes y frecuencias), la digitalización implica discretizar esta señal de dos dimensiones (tiempo, amplitud) a una frecuencia de muestreo, por lo que trabajaremos con un vector unidimensional con los valores de las muestras de amplitud ordenadas. Para caracterizar señales de audio existe una gran cantidad de indicadores, como su energía, amplitud media, ancho de banda, frecuencia fundamental, o diversos tipos de coeficientes como los ceptrales y los MFCC[2]. A partir de estos se pueden utilizar diversas técnicas de aprendizaje automático con un enfoque clásico como *Clustering*, *PCA*, *Decision Trees* o ajuste de gaussianas. Para que el aprendizaje sea exitoso se requiere que la especificidad acústica de las cotorras sea suficiente para que las características se diferencien de los demás sonidos, lo cual parece completamente viable considerando [4, 5, 6, 7], ya que si se diferencian individuos dentro de la misma especie, aún más factible es diferenciarlas de otras especies (o fuentes de sonidos).

Se realiza un relevamiento de las principales técnicas y modelos utilizados para procesar el audio de las señales producidas por las cotorras. En [6] (pgs. 3 y 4) y [5], donde se estudia la variación de *contact calls* en poblaciones migrantes (Uruguay - Texas,US), se extraen 203 indicadores combinando los métodos anteriormente mencionados más algunos indicadores acústicos estándar (como son la media de frecuencias, desviación estándar de frecuencias, primer cuartil de frecuencias, entropía espectral, *spectral flatness*, etc). Los modelos de aprendizaje que se utilizan son *Supervised Random Forest* y *Gradient boosting*. También es frecuente el análisis de la *Spectrographic cross correlation* (SPCC)[1] y los *Mel-Frequency Ceptral Coefficients* (MFCC)[2], que aparecen en varios artículos como método para medir similitud entre señales [4].

Aprendizaje profundo

Pese a los buenos resultados de los métodos mencionados, en los últimos años se ha vuelto cada vez más frecuente el análisis digital de las señales de audio a partir de su espectrograma. El espectrograma se construye como la sucesión ordenada (representando el transcurso temporal) del espectro de frecuencias calculado sobre

[1] La SPCC es una medida de similitud entre dos señales, como correlación entre sus espectrogramas [20]

[2] Los MFCC se calculan segmentando el audio en pequeñas ventanas a las que se les aplica la DFT para obtener la potencia espectral, y luego se los lleva a la escala Mel [21], construida en base a la percepción humana de los tonos. Por ser coeficientes basados en la percepción auditiva humana, son muy utilizados en el reconocimiento de voz.

ventanas de tiempo de la señal. Como el espectro es bidimensional, el espectrograma resulta en una gráfica tridimensional que representa la energía del contenido frecuencial de la señal en el transcurso del tiempo, llevado a dos dimensiones mediante una escala de colores que representa la intensidad para cada punto (frecuencia, tiempo). Dado que los indicadores previamente mencionados se basan en su mayoría en propiedades espectrales y de energía, dicha información está en gran medida presente en el espectrograma. A partir del ingente éxito del procesamiento de imágenes con arquitecturas de redes neuronales profundas (en particular las redes convolucionales), y dado que el espectrograma es al fin y al cabo una imagen, se ha encontrado una gran predominancia de este método en los sistemas modernos de clasificación de sonidos.

3.2. Linea de trabajo: clasificación de espectrogramas

Dada la predominancia bibliográfica [12, 13], sus buenos resultados, y considerando que los métodos clásicos pueden implicar un nivel de ajuste o precisión innecesario (e inviable) sobre las propiedades acústicas, para esta aplicación parece más que suficiente comenzar por la clasificación de imágenes de espectrograma. A su vez, gran parte de la literatura reciente sobre reconocimiento de aves por sonido utiliza arquitecturas de CNN con espectrograma como entrada [14, 16, 17], y en algún caso los MFCC [15]. Entre las investigaciones al respecto, se considera de gran relevancia el proyecto BirdNET (sumado a la app eBird) del Cornell Lab of Ornithology, que consiste en una gran base de grabaciones de aves construida de manera abierta por los usuarios, capaz de reconocer cualquier ave capturando unos pocos segundos de espectrograma, utilizando CNN [17, 18, 19].

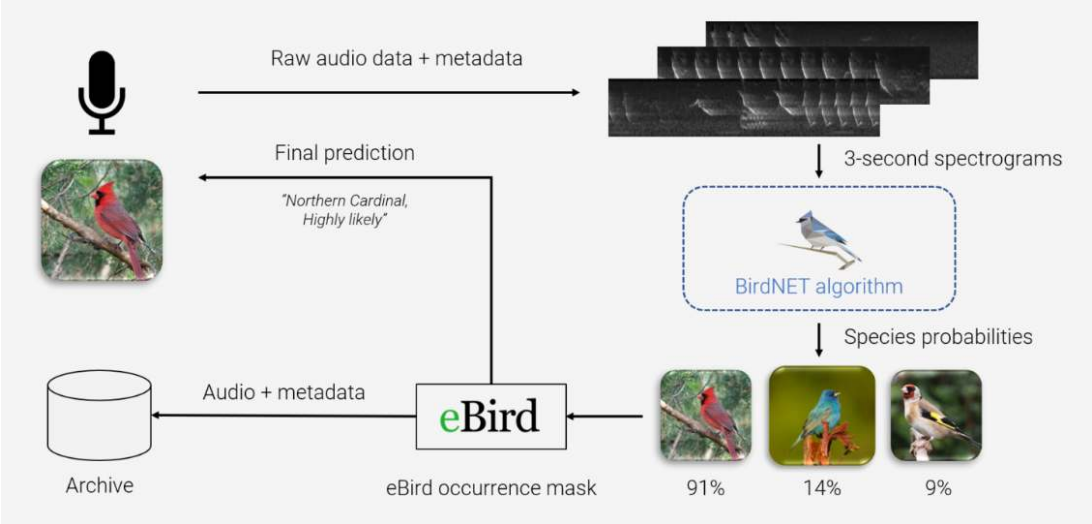


Figura 3: Esquema del modelo de clasificación de BirdNET (birdnet.cornell.edu) [17]

3.3. Primera limitante

Sin embargo, existe una limitante a considerar: el costo computacional y peso del modelo a utilizar. Es preciso contemplar que el Hardware (HW) debe procesar los datos provenientes de varios micrófonos de manera simultánea y en tiempo real, y forma parte de un mismo sistema embebido en el campo. Esto presenta una limitante de procesamiento y de consumo que ha de tenerse en cuenta desde el punto de partida. No obstante, se han encontrado varias aplicaciones de detección y clasificación de eventos acústicos bajo el paradigma *TinyML*, conformados por un HW sencillo como raspberry pi o arduino, y un modelo de clasificación acorde basados principalmente en redes convolucionales con imágenes espectrales de entrada [22, 23, 24]. Por tanto, parece viable continuar en la linea de modelos de visión, teniendo en cuenta este equilibrio entre la calidad de imagen, o sea la resolución del espectrograma (que se traduce en tamaño de la entrada), capacidad del modelo para ajustarse a los datos (que implica una cierta cantidad de parámetros), y la viabilidad del modelo considerando las limitantes de HW. Notar que el tamaño de las redes convolucionales para imágenes escalan con mucha facilidad.

En consideración de estos puntos, se procedió con el espectrograma como señal a procesar. El análisis mediante MFCC, muy utilizado en reconocimiento de voz humana, en un principio está descartado dado que ofrece una sensibilidad capaz de distinguir palabras, o un hablante de otro, que en este caso no es requerida. El

sistema debe tener suficiente robustez en vista del ruido que puede presentar el medio exterior, y esto parece factible por la vía del espectrograma.

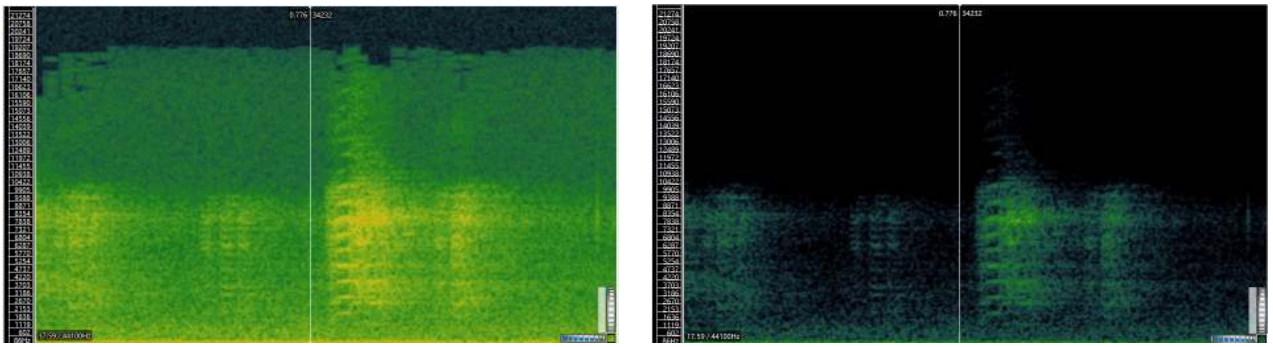
4. Los datos

Dado que es un problema de clasificación binaria, se necesitan pistas de audio que representen ambas clases: cotorras y ruido de ambiente. Los audios de cotorras deben incluir los diversos llamados que emiten y una cierta variabilidad en el número de cotorras emisoras, de tal manera que funcione tanto para una cotorra, un grupo chico o una bandada. El ruido de ambiente debe representar todos los posibles tipos de sonidos en el campo, incluyendo de manera particular otros pájaros o animales. La actual base de datos está constituida de audios de diversas fuentes públicas extraídas de internet, como grabaciones y videos, más varios audios de autoría propia. La misma contempla: **ruido de campo (Uruguay, monte), trigo y viento, viento y pradera, viento y arboles/hojas, lluvia, lluvia en el campo, cortadora de pasto, tractor, grillos, chicharra, personas y pájaros**. Las especies de aves incluidas son: **benteveo, carancho, cardenal, carpintero, gallina, gavilan de campo, horneros, pirincho y tero**.

Por otra parte, las grabaciones de cotorra provienen de videos en la web, gran cantidad de grabaciones de autoría propia (de diversas localidades del país grabados en el verano de 2024 mediante la aplicación de eBird), y se apartaron unicamente para *testear* por fuera del conjutno de validación algunas pistas obtenidas de grabaciones de usuarios en la página web de eBird [17].

4.1. El espectrograma de las cotorras

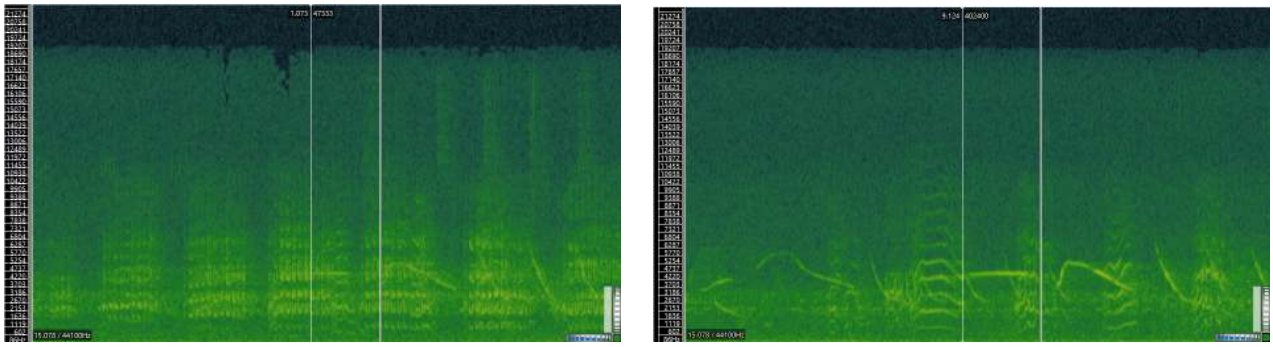
Con las herramientas de software Sonic Visualiser y Audacity, se estudia el espectrograma de las cotorras a fin de ajustar los parámetros de construcción del mismo que más nítidamente expresen los patrones propios de los sonidos emitidos por las cotorras (sus armónicos), que son el tamaño de la ventana (en número de muestras), el porcentaje de solapamiento y la escala. Se fija la ventana en 512 muestras y 87.5 % de solapamiento, como se muestra en el siguiente ejemplo, que abarca aproximadamente un segundo de audio, con un fuerte gorgoteo de cotorra.



(a) Escala de energía en dB (b) Escala en dB^2

Figura 4: Capturas de Sonic Visualiser (Window 512, 87.5 %), aprox. un segundo de duración

En las figuras 5 se observan poco más de dos segundos de audio, donde la imagen izquierda muestra sonido emitido por cotorras, y el de la derecha otra especie, con un canto más nítido y constante. El patrón de la cotorra es una constante en todas las pistas de sonido observadas, con gorgoteos muy veloces intermitentes con breves pausas entre ellos, y un barrido muy amplio de frecuencias propio del carraspeo que realizan. En la figura 6 se observa la misma pista de audio, con presencia de cotorra en los primeros segundos, y de la otra especie en los últimos. En la sección 6.4 se profundiza sobre la forma del espectrograma para las cotorras.



(a) cotorra (b) otra especie

Figura 5: Capturas de Sonic Visualiser (Espectrograma: Window 512, 87.5 %, dB), aprox. dos segundos de duración

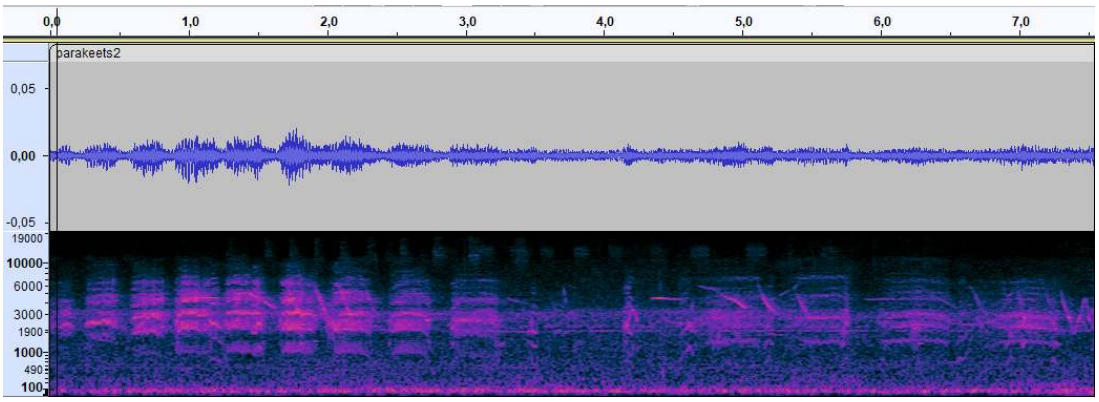


Figura 6: Captura de Audacity con forma de onda y espectrograma

A partir de estas observaciones, y en vista de algunas de las aplicaciones citadas anteriormente, parece razonable procesar el audio de a muestras de cinco segundos de duración (o tal vez tres), de tal manera que las imágenes de espectrogramas con las que se entrena la red neuronal sean constantes en sus escalas y en los parámetros de construcción del mismo. En general, los casos relevados que utilizan este método varían entre el segundo y los diez segundos de muestra; en particular, el modelo de la aplicación eBird [17] espera tres segundos de entrada. Muestras mayores a cinco segundos devienen en pérdida de detalles importantes sobre los armónicos principales, y si bien una pista más corta (de por ejemplo un segundo) puede hacer más nítido el patrón fundamental, pierde la información de 'mayor orden' en lo que refiere a los periodos de espaciamiento entre un gorgoteo y el siguiente, lo que a su vez es mucho más fácil de capturar por un micrófono en condiciones de ambiente ruidoso. Sumado a esto, acortar las muestras de audio también implica multiplicar el costo computacional y tiempo de procesamiento.

4.2. Conjunto de datos

En suma, el *set* de datos resulta de la siguiente manera, apartando un 30 % del mismo para validación de forma aleatoria (siendo True la etiqueta de 'cotorra'):

- Conjunto de Train: 511 datos False (43.6 minutos de grabación) / 309 datos True (26.75 minutos).
- Conjunto de Validación (30 %): 219 datos False (18.3 minutos de grabación) / 132 datos True (11 minutos).
- Total: 1171 muestras de 5 segundo (98 minutos).

5. Modelo de clasificación: Red neuronal convolucional (CNN)

5.1. Arquitectura

A partir de la arquitectura sugerida en [12], luego de muchas pruebas de desempeño modificando diversos parámetros, atendiendo al *trade off* entre el tamaño y peso del modelo y su performance como clasificador, la estructura del modelo se resume en el esquema de la figura 7.

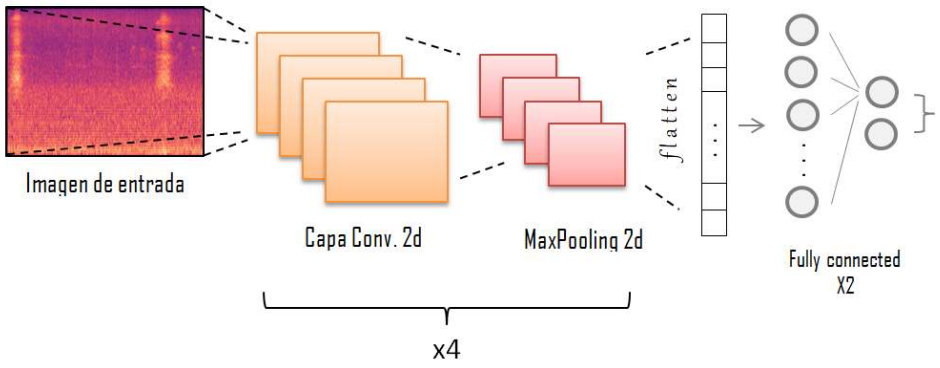


Figura 7: Diagrama de la estructura de la CNN modelada

Notar que esta arquitectura, para una entrada de tamaño $192x256$ (que se discute en la sección 5.2), cuenta con una cantidad de parámetros entrenables en el orden de los 700 mil, y ocupa un espacio de 2.64 MB, que si bien no es despreciable, es relativamente bajo para los estándares de una CNN. Se observa que para acotar el número de parámetros lo más efectivo es regular la capa densa *fully connected*, mientras que es apropiado conservar cierta profundidad en la cantidad de capas convolucionales, a fin de aprovechar su capacidad de extracción de características ópticas.

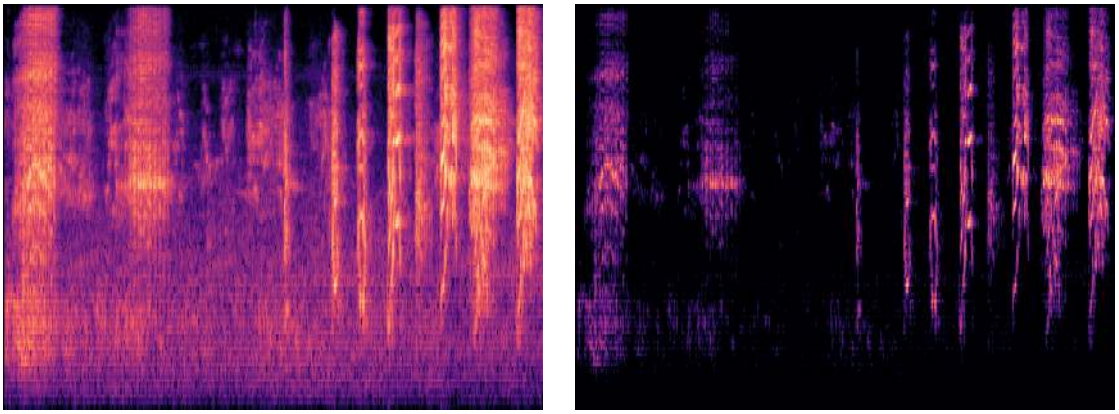
Las primeras variables a considerar fueron la propia estructura, donde en primera instancia se constata que es necesario mantener un número de parámetros considerable para que la red neuronal pueda ajustarse a las características de las imágenes, de lo contrario el modelo se subajusta. Por ejemplo, para este set de datos, si la penúltima capa densa baja de las diez neuronas, la salida es siempre cero (clasificación negativa). Aunque se desee cambiar de estructura, se debe tener en cuenta que el problema de interpretación de los espectrogramas exige un modelo con cierta complejidad.

5.2. Preprocesamiento de los datos de entrada

Luego se analizó la conveniencia del tamaño de entrada de las imágenes. Si bien este parámetro es el de mayor incidencia en el costo computacional, dado que escala en forma cuadrática y se multiplica por tres canales (RGB), disminuir el tamaño de la imagen deteriora la resolución del espectrograma. Se estima las dimensiones 196x256 (conservando las proporciones de la imagen original) píxeles como un tamaño suficiente para un buen desempeño, puesto que para este set de datos el *accuracy* en validación supera el 86 %, lo que es muy auspicioso. Como la performance es muy buena, y este reporte no se realiza partiendo de un Hardware determinado, se prefirió ser conservador en el uso de recursos. Sin embargo, si las capacidades de cómputo lo permiten, es conveniente expandirlo para evitar la pérdida de detalle. En este trabajo se buscará una solución intermedia.

Otra técnica utilizada fue crear los espectrogramas con escala de energía en dB^2, que acentúa las diferencias entre valores de energía bajos y altos (umbralizando los bajos), como se observa en la figuras 4 y 8. Sin embargo, aunque el desempeño haya sido bueno, no mejoró el *accuracy*, por lo que se decidió continuar con escala en decibelios.

Por otro lado, también se consideró filtrar con un pasa banda (en el rango frecuencial de actividad de las cotorras) todos los audios, previa creación del espectrograma, de tal manera que la imagen se centre en la zona de interés. Esto podría ser útil si se contara con un tamaño estimado fijo de imágenes, y este fuera muy acotado (por las exigencias de HW, comunicación, etc.), donde sea necesario quitar regiones inactivas en pro de no perder detalle para el sonido de las cotorras. Pero como no es el caso, carece de sentido quitar información que el modelo también aprovecha, dado que las frecuencias de menor (o nula) energía también son información para determinar si es o no es cotorra.



(a) Energía en dB (b) Energía en dB^2

Figura 8: Ejemplo de espectrograma (grabación con presencia de cotorras - cambio de escala)

Determinados estos puntos, partiendo de las dimensiones de imagen mencionadas, se realizaron varias pruebas variando distintos parámetros de entrenamiento y de la arquitectura, procurando mantener un tamaño acotado. En el cuadro 1 se adjuntan los modelos con mejores resultados.

Modelo	*input_shape*	Número de Parámetros	Espacio	Acc. % (Train)	Acc. % (Val.)
ModeloSimple	192x256	691518	2.64 MB	85.75	86.65
ModeloHD	240x320	932158	3.56 MB	98.29	92.61
ModeloHD+DataAug	240x320	932158	3.56 MB	99.89	94.32
(TL + dense 10,2)	240x320	3247904	12.39 MB	100	98.01

Cuadro 1: Mejores experimentos

Para todos los experimentos registrados en el cuadro (menos el último), la arquitectura consta de cuatro capas convolucionales y dos capas densas (20 y 2 neuronas), optimizador Adam, *binary crossentropy* como función de pérdida, *batch size* de ocho (por limitaciones de SW), y doce épocas de entrenamiento. Si bien el primer experimento con el tamaño de entrada base (192×256) es bueno, se comprueba que subir un tanto las dimensiones de la imagen de entrada produce un salto cualitativo en el *accuracy* de validación, y el modelo continúa siendo más bien chico.

Se reentrena el modelo para imágenes de 240×320 luego de realizar *Data Augmentation* sobre el conjunto de entrenamiento, con el criterio explicado en la sección 5.3. Esta medida aumenta dos puntos el *Accuracy* en validación, que resulta muy positivo si se considera que el tamaño del modelo permanece igual. En la figura 9 se adjunta un ejemplo de inferencia para un espectrograma con presencia de cotorra, y la salida de la imagen luego de la capa *max pooling* correspondiente al primer filtro de convolución. Notar que se genera un mapa de calor que acentúa la zona de interés, suavizando la información menos importante.

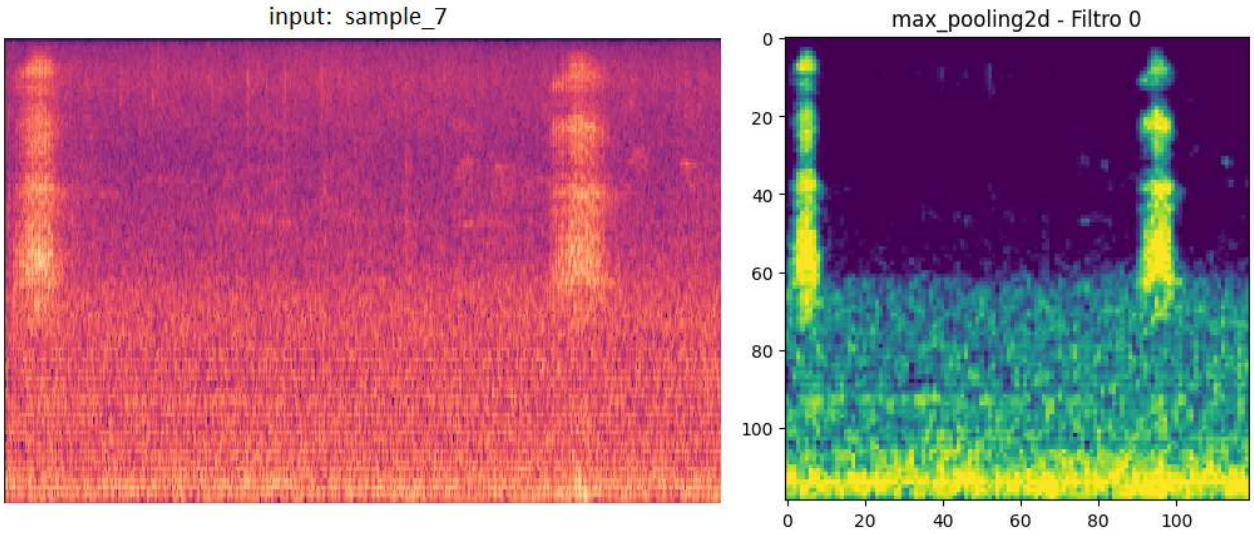


Figura 9: Espectrograma (audio con sonido de cotorra) luego de la primer capa convolucional

5.3. Data Augmentation

A un 30 % de la base de cotorras (*label = True*) del conjunto de entrenamiento se le aplica *Data Augmentation*, de tal manera que las clases queden más balanceadas (mientras que el conjunto de validación se mantiene exactamente igual).

- A un 15 % se le suma Ruido Blanco en dos niveles de potencia (moderados).
- Al otro 15 % se le suma ruido natural proveniente de nueve muestras de *backround* (campo, lluvia, grillos, algunos sonidos de pájaro, etc.).

El resultado fue aumentar la muestra de cotorras en Train de 309 a 402, representando un 44 % del total del conjunto de Train, frente al 38 % previo al aumento. No se realizó *Data Augmentation* variando velocidad, volúmen o frecuencias (traslación) por no resultar representativo de la realidad.

5.4. Transfer Learning

Finalmente se realiza un experimento para estimar el desempeño de utilizar modelos preentrenados. Como un ejemplo, a sugerencia de [12], se utiliza la red MobileNetV2, CNN preentrenada de Google, optimizada para dispositivos móbiles, con un modelo chico que requiere bajo poder de cómputo y poca memoria. Se utiliza como preprocesamiento de las imágenes de entrada, para obtener sus *embeddings*, y luego entrenar una *fully conected* de dos capas con los vectores de características extraidos por dicho modelo. El resultado fue un *accuracy* en validación del 98 %.

Lo más preciso hubiese sido utilizar modelos preentrenados con espectrogramas para clasificación de audio, no obstante, hay fuertes evidencias que sugieren que redes pre-entrenadas en ámbitos distintos al espectrograma, logran transferir características útiles aprendidas, como señala [25]. En dicho artículo se prueba que *ImageNet-Pretrained standard deep CNN models* constituyen una base fuerte sobre la cual continuar entrenando con espectrogramas para clasificación de audio. Sin embargo, en tal escenario el tamaño del modelo se expande considerablemente.

5.5. Modelo final: últimas consideraciones

Si bien el mejor resultado se obtuvo utilizando un modelo preentrenado, se decidió conservar "ModeloHd+DataAug" (o sea imágenes de 240×320, entrenado con *Data Augmentation*) como modelo a utilizar, ponderando la relación desempeño - tamaño del modelo. No es el objetivo de este trabajo obtener el mejor modelo posible, si no esbozar las mejores líneas como parte de un estudio de viabilidad. Lo óptimo sería volver a entrenar un modelo de similares características, e incluso entrenar sobre el modelo que proponemos, con una base de datos que contemple el ambiente donde se va a instalar el sistema, y abarque más y mejores grabaciones de cotorras. En particular, se considera de gran interés distinguir entre los distintos tipos de llamados, para que

el conjunto de datos esté bien representado en este sentido, y muy especialmente grabar cotorras en el propio campo de aplicación cuando bajan a comer. Allí reside la principal debilidad del modelo.

En la figura 10 se adjunta la matriz de confusión en el conjunto de validación para el modelo seleccionado. Notar que presenta un tanto más de tendencia al Falso Positivo que al Falso Negativo, lo que parece bueno, a los efecto de este problema. Parece mejor equivocarse al enviar el dron aunque no hayan cotorras, a no enviarlo y que estas continúen dañando los cultivos.

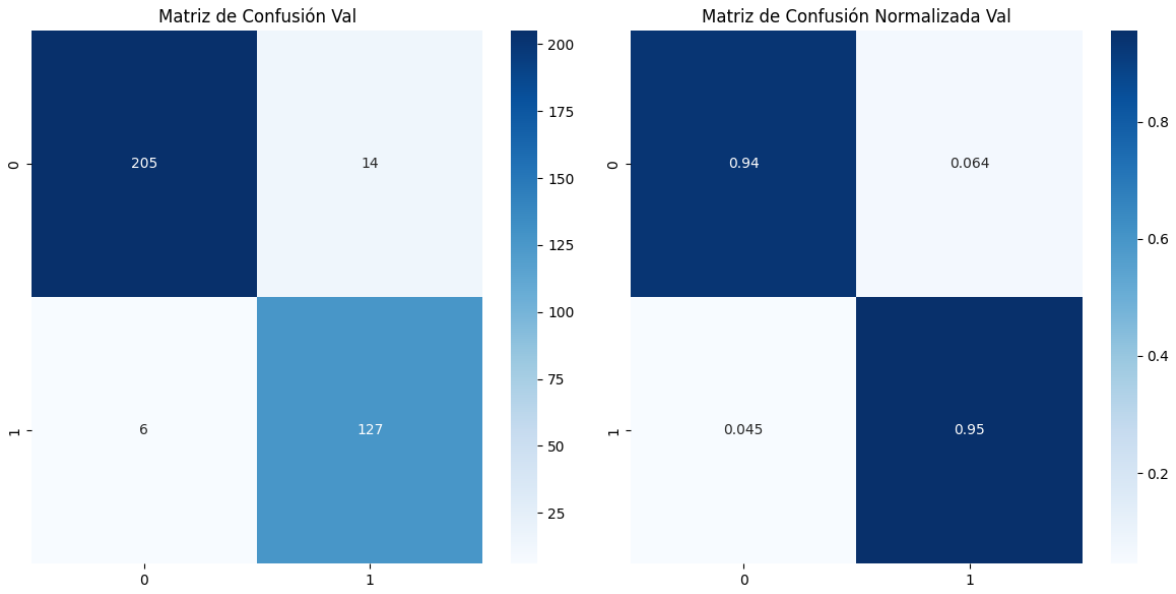


Figura 10

6. Detector de eventos

Dado que el clasificador binario espera entradas de cinco segundos para determinar la presencia o no de cotorras, no parece viable calcular el espectrograma y procesarlo como imagen en la red neuronal cada cinco segundo de forma permanente, menos aún considerando que el sensado proviene de una red de varios micrófonos. Por esto, se hace necesario un detector de eventos como instancia previa de filtrado, que mantenga un costo computacional (y de consumo) bajo, pero que ante un evento acústico relevante comience a enviar fragmentos de cinco segundos al clasificador para determinar si se trata de una cotorra.

6.1. Dinámica de sensado: ventana deslizante

Esencialmente los requisitos son dos: mantener simples las operaciones de detección, y obtener un indicador que dé positivo para todos los casos donde existan cotorras (no haya falsos negativos), aunque implique detectar eventos sin cotorra (falso positivo), procurando mantenerlos al mínimo. Para esto se plantea un sistema que cada una cantidad de tiempo T, realiza una operación con resultado R sobre la señal en la última ventana de tiempo $W > T$, y si R_n dista un umbral u de R_{n-1}: hay evento.

Los indicadores evidentes son la energía de la señal y la amplitud promedio (análogos), que se calculan:

$$E_{W_n} = \sum_{W_n} x^2[n]$$

$$RMS_{W_n} = \sqrt{\frac{1}{N}\sum_{W_n} x^2[n]} = \sqrt{\frac{E}{N}}$$

Si (por ejemplo):

$$E_{W_n} > E_{W_{n-1}} * u \rightarrow Evento! \tag{1}$$

Donde x[n] es la señal en el tiempo, y en este caso tomamos u como un umbral relativo, por ejemplo 2, lo que implica que la energía debe duplicarse de una ventana a la siguiente.

6.2. Ensayos sobre base sintética

Con el fin de ensayar y ajustar el detector de eventos, se tomaron algunas pistas (y crearon otras) que contemplen un conjunto de casos que abarquen las posibilidades que pueden ocurrir en la práctica. La idea es que el 'ruido de fondo' típico del campo no active el detector, y sí lo hagan sonidos que se alejen de la media (como puede ser un pájaro, un ladrido, el sonido de una máquina, etc.). El concepto de ventana deslizante, y por tanto media móvil, procura una adaptación a las condiciones acústicas (variables) de base en un determinado momento y lugar, además de permitir cambios graduales y/o 'suaves', pero reconocer cambios bruscos.

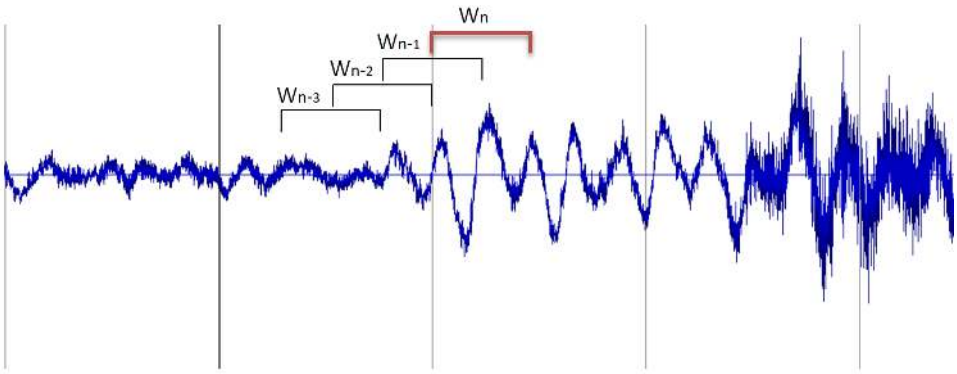
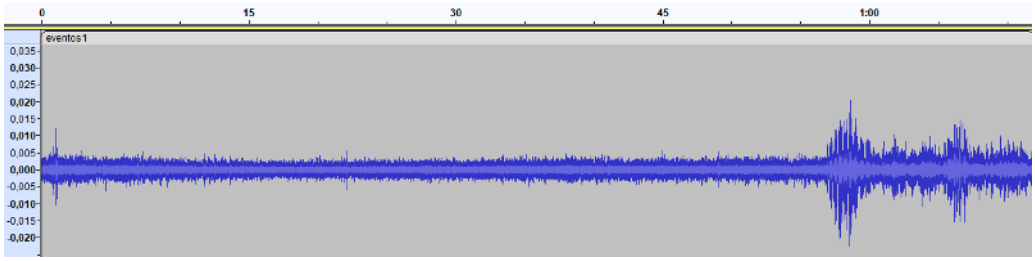


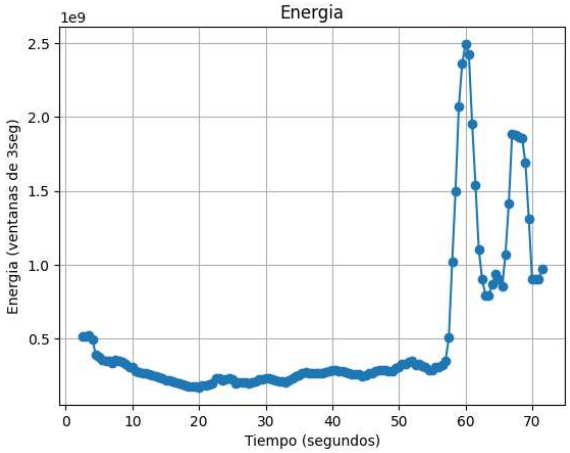
Figura 11: Gráfico de ventana deslizante

La figura 11 muestra gráficamente el concepto de la ventana deslizante, donde a partir de W_n el cambio en la operación calculada respecto a la ventana anterior es tal que activa el detector de eventos. Naturalmente, se podría considerar k ventanas anteriores ($[W_{n-1}...W_{n-k}]$), mediante una selección de coeficientes que ponderen el peso relativo de las ventanas pasadas, y así contar con mayor memoria en el proceso y un cálculo más preciso. Sin embargo, dada la variabilidad del problema, se busca un sistema más bien robusto y genérico; mientras que aumentar los parámetros implica ajustar el detector a los casos de ensayo. Por esto, se opta por la versión más sencilla, i.e: memoria de una sola ventana, ponderada por un solo umbral relativo (porcentual).

En la figura 12 se observa un ejemplo de aplicación del algoritmo 1, donde la energía se mantiene bastante constante de a ventanas de tres segundos (medidas con una frecuencia de medio segundo) durante un ruido de viento estable, y aumenta sustancialmente cuando un conjunto de cotorras comienzan a comunicarse en el segundo 57. Notar que el salto cerca del segundo 57 es muy importante, por lo cual, con un umbral $u = 1{,}5$ el evento es detectado con éxito por sobrepasar en más de un 50 % el valor de Energía de la ventana anterior.



(a) Señal en el tiempo, captura del Audacity



(b) Gráfico correspondiente para c./ventana de tiempo

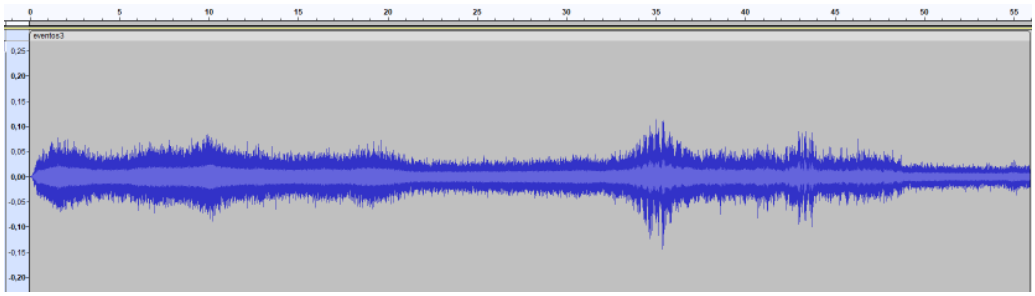
Figura 12

6.3. Casos problemáticos y discusión

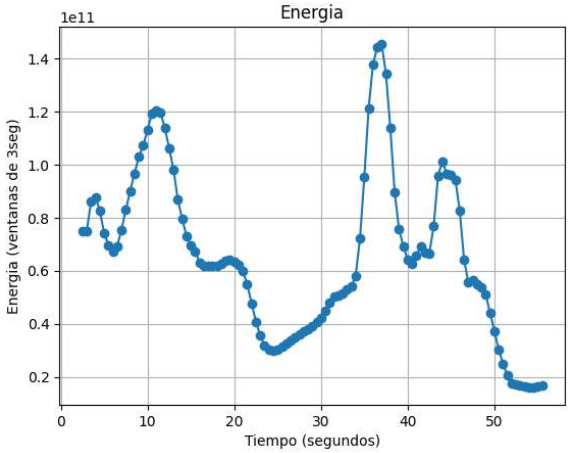
A pesar de que se obtienen resultados prometedores para casos sencillos como el anterior, existen situaciones nada inusuales, donde un aumento de ruido (por ejemplo de viento) hace incrementar la energía de la señal considerablemente hasta picos equiparables a los producidos con presencia de aves, como en la figura 13. Para este caso, en los primeros 10 segundos ocurre un incremento del ruido, que produce una variación en el gráfico de energía, pero recién en el segundo 33 comienza el evento como tal (sonido de aves). No obstante, para este caso es notorio cómo en el pico correspondiente al evento los saltos de energía entre una ventana y la siguiente son mayores, mientras que con el ruido el aumento es más gradual, a pesar de que se alcancen niveles de energía similares.

Esto podría resolverse aumentando el umbral de eventos (porcentaje de la medida en la ventana anterior), sin embargo, un umbral más estricto que filtre los aumentos de ruido puede eventualmente ignorar eventos, por lo que existe un nuevo compromiso de sensibilidad. Si bien puede resolverse mediante entrenamiento, en la práctica se observa que eventos acústicos claramente distinguibles al oído humano exigen umbrales bastante sensibles, al igual que incrementos de ruido activan el detector según los parámetros ajustados con los casos positivos. Surgen ejemplos como el de la figura 14, contrarios al caso 13, donde el ruido emitido por las cotorras comienza en el segundo 9, y es suficientemente perceptible, pero no es hasta el segundo 14 donde el salto de energía es cualitativo.

Parece que dadas las condiciones de ruido esperables, no es viable hacer uso únicamente de la energía de la señal, y se hace ineludible explotar la información que deriva de la forma de onda, o bien, la frecuencia. Esto introduce un problema fundamental, dado que el análisis espectral y los indicadores derivados demandan una cantidad de cálculos (costo computacional) mayor, lo cual era necesario evitar en esta instancia. Una idea sencilla fue contabilizar los cruces por cero, pero se comprobó que no guardan una correlación necesaria con la frecuencia.

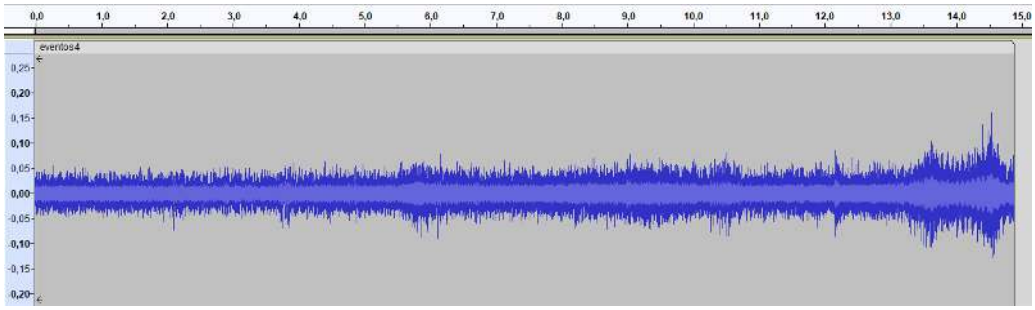


(a) Señal en el tiempo, captura del Audacity

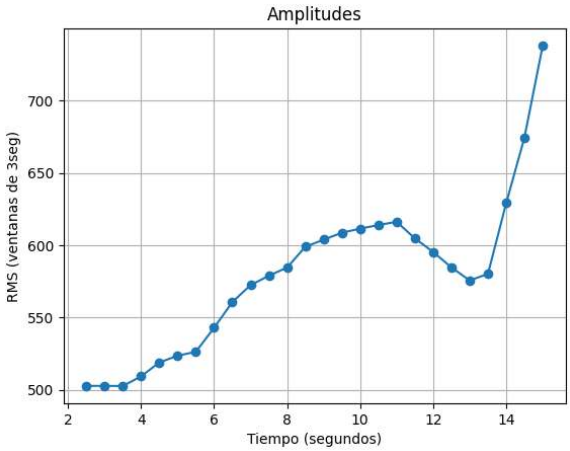


(b) Gráfico correspondiente para c./ventana de tiempo

Figura 13



(a) Señal en el tiempo, captura del Audacity



(b) Gráfico correspondiente para c./ventana de tiempo

Figura 14

6.4. Filtro pasa banda

Finalmente, se presenta como mejor alternativa acotar la medida de energía de cada ventana temporal a las componentes de la señal en un rango acotado de frecuencias. Si la banda de frecuencias comprende las componentes más fuertes del espectro de la cotorra, aunque en dicha franja exista ruido, el peso relativo de la energía del ruido en relación a la cotorra disminuye enormemente, haciendo viable el algoritmo anterior basado en umbral de energías. De hecho, en la práctica se comprueba que el ruido suele 'distribuirse' en un amplio espectro de frecuencias.

En cuanto al costo de esta medida, si bien un filtro por frecuencias vía Software implica realizar cálculos 'pesados' (a estos efectos), como la FFT para hallar el espectro, o análogos, y descomponer la señal (digital) en el tiempo, no así un pasa banda electrónico que filtre la señal analógica previamente. Por vía de Hardware la medida es altamente viable.

Elección de la banda de frecuencias

Analizando nuevamente el espectro de las cotorras, como se observa en la figura 15, se selecciona inicialmente la banda $[2000, 7500]Hz$, con el ánimo de procurar un filtro tendiente a la flexibilidad, no exhaustivo, donde el falso negativo es menos deseable que el falso positivo. Con esta franja buscan contemplarse todos los llamados de las cotorras, y se considera satisfactorio que otras aves activan el evento, sin embargo, de ser necesario puede acotarse aún más. El análisis (al igual que en la sección 4.1) se contrastó con la información proporcionada en las investigaciones citadas, en particular, se adjunta la figura 16, con una serie de gráficos perteneciente a [5]. Se observan componentes principales en frecuencia muy análogas, con mayor energía en la banda de frecuencias en cuestión.

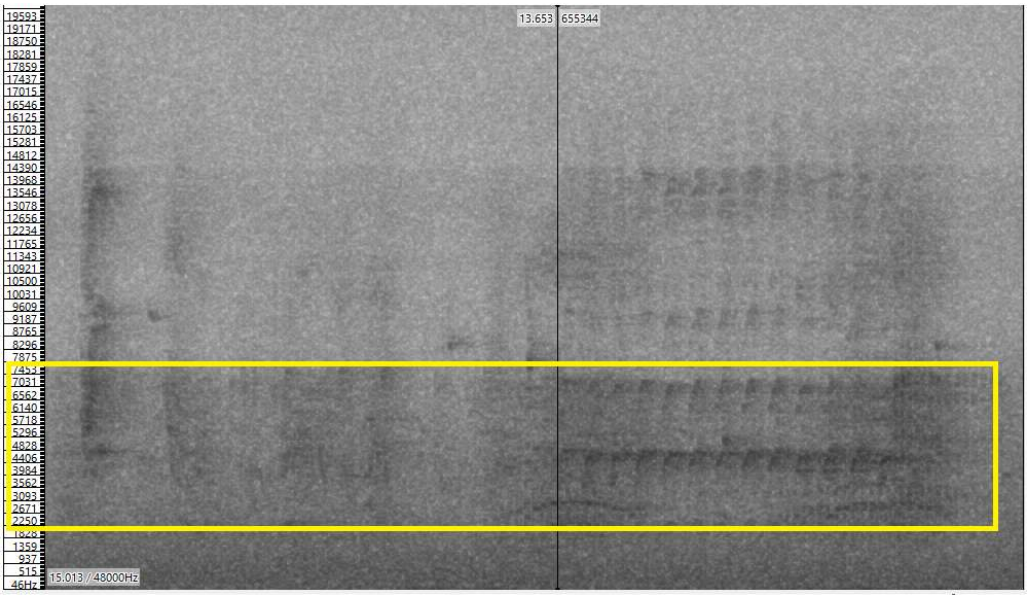


Figura 15: Captura de Sonic Visualiser del espectro de cotorra, encuadrada la franja $[2 : 7{,}5]kHz$

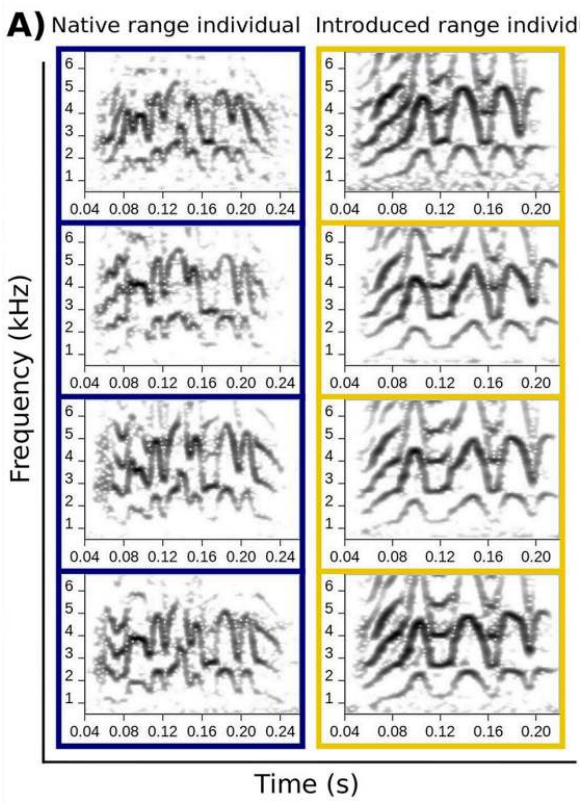


Figura 16: Extraido de [5]: 'Native and introduced range monk parakeets displayed strong individual vocal signatures. . In A) we show a lexicon with 4 contact calls for one repeatedly sampled bird in each of the native and introduced ranges'

Desempeño del detector con filtro pasa banda

El desempeño mejora notablemente, haciendo viable el algoritmo planteado. De un par de decenas de audios sintéticos, logra detectar con éxito todos los eventos. En la figura 17 se adjunta el ejemplo problemático tratado previamente (figura 13), donde el objetivo es obviar el primer tramo de incremento de energía (por el viento), y activarse con el segundo y tercer pico, producidos en el rango de frecuencia correspondiente a las cotorras.

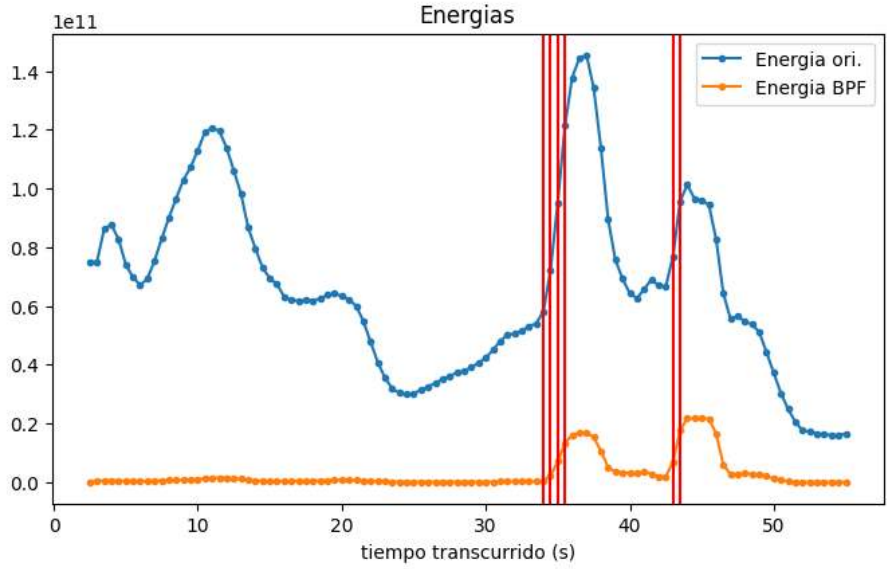


Figura 17: Gráfico de energía (con y sin filtro pasa-banda) para c./ventana de tiempo. Linea roja: evento

Notar que la curva naranja representa para cada instante la medida de energía correspondiente a los últimos tres segundos aplicando el filtro pasa banda, la curva azul la energía original sin filtro y la linea roja los instantes donde hay detección de eventos según el algoritmo 1. Luego de varias pruebas se fija el umbral porcentual en un 60 %, aunque es conveniente adaptarlo a las condiciones donde se instale el sistema.

Respecto al ejemplo de la figura 17, en el segundo 33 comienza el sonido de cotorras, disminuye y en el segundo 44 reanuda. Por otro lado, para el otro caso comentado en la sección de casos problemáticos (figura 14), donde el objetivo es distinguir el sonido de las cotorras que comienza en el segundo 9, en medio del ruido, y vuelve a aumentar en el 14, el sistema con pasa banda también funciona exitosamente (figura 18).

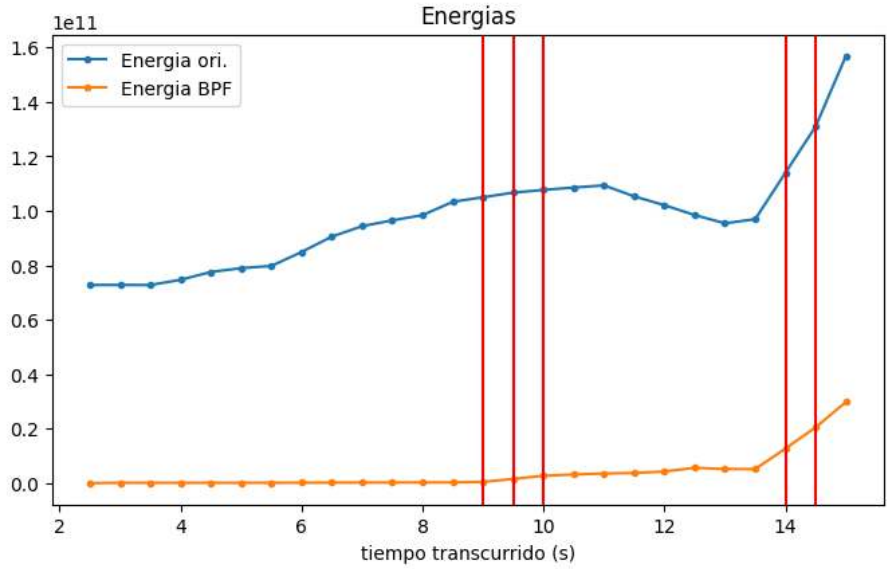


Figura 18: Gráfico de energía (con y sin filtro pasa-banda) para c./ventana de tiempo

El filtro pasa banda cumple con el cometido de mantener la energía de la señal estable cuando no hay presencia de cotorras, y aumentar cuando sí la hay. El único problema detectado que aún persiste es el escenario donde el ruido es muy bajo (cercano a cero), y cualquier aumento del ruido en el rango de frecuencias de interés es suficiente para superar con creces el 60 % de la energía anterior, activando el detector. Sin embargo, como antes se ha mencionado, es preferible la existencia de falsos positivos a falsos negativos. Sería un problema si el ruido estuviese constantemente variando entre valores muy bajos y aumentos periódicos. De ser el caso, este problema podría solucionarse adicionando un umbral no porcentual, donde el evento requiera de un amento de la energía anterior y de estar por encima de cierto valor de energía razonable.

También puede ocurrir lo contrario, que el nivel de energía en la banda de frecuencia sea muy alto, por las características del ruido, y que no llegue a aumentar un 60 % al ocurrir un evento. En tal caso parece viable flexibilizar el umbral, por ejemplo en un 40 %, y/o ajustar el rango de frecuencias a una ventana más chica, donde el sonido de las cotorras adquiera aún más predominancia.

7. Sistema completo

El sistema completo consistiría en el detector de eventos, que funciona de manera constante, y cuando ocurre un evento, envía los últimos 5 segundos al clasificador. El clasificador convierte la ventana temporal en una imagen a partir del cálculo del espectrograma, adapta su tamaño y lo ingresa en la red neuronal entrenada. Si la clasificación es positiva se envía el dron, se detiene el detector un tiempo prudente para ahorrar consumo, se espera una segunda detección como confirmación, o el flujo que se desee en la implementación del sistema. Para probar el *pipeline* completo, se realizó una simulación del sistema de tal manera que emule el comportamiento como si las muestras llegasen en tiempo real.

En la figura 19 se observa un ejemplo, que corresponde a un audio que contiene el sonido de una cotorra y de un benteveo, donde ambos activan el detector de eventos, pero son clasificados de manera distinta. Las lineas verticales en el gráfico marcan el tiempo donde se detectó eventos; si la clasificación es positiva la linea es verde, si no, roja. Bajo el gráfico, el espectrograma correspondiente. En base a esto, ambas clasifiaciones son correctas.

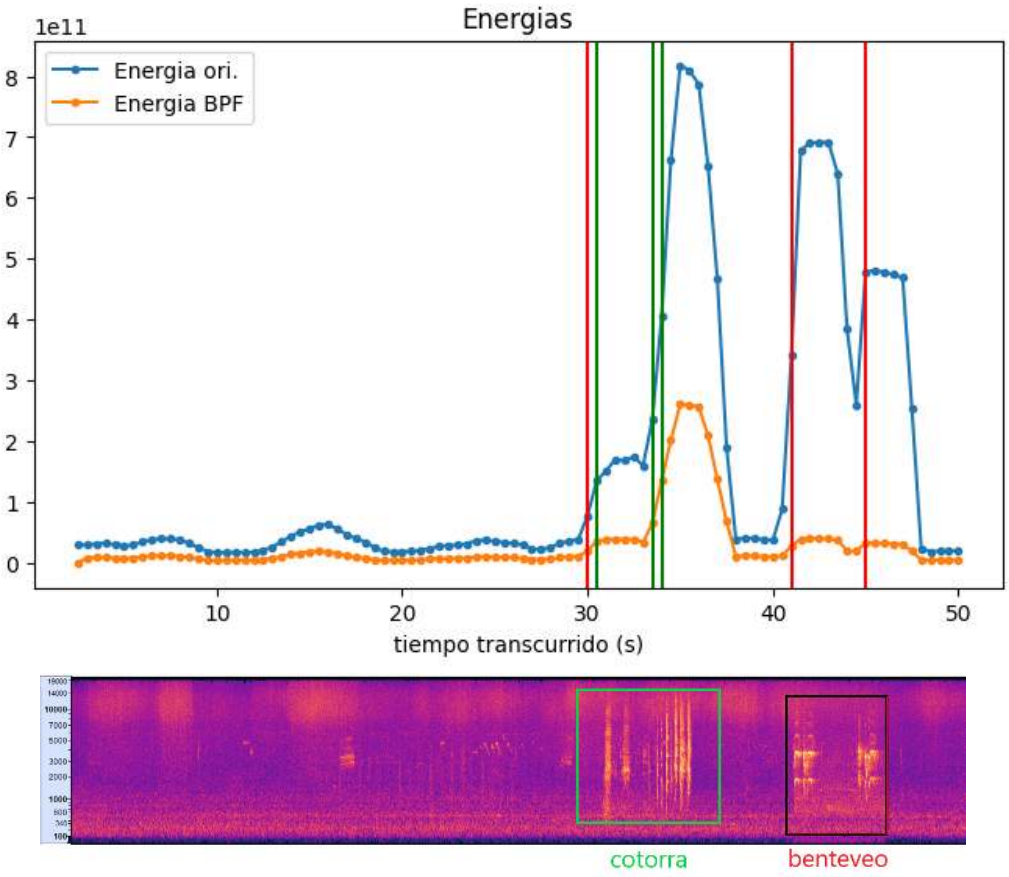


Figura 19: Resultado gráfico de la simulación

En este caso la muestra dura 50 segundos, pero en tiempo real la alimentación de audio es constante. Por otro lado, en la implementación real se espera que el procesamiento de la imagen de espectrograma para clasificación se realice de forma paralela, mientras el detector continúa. Notar que si la clasificación durase más de 0,5 segundos (no es el caso), podría producirse una cola de espectrogramas para clasificar. En el código que implementa esta simulación, las funciones son en serie, por lo que se interrumpe unos instantes el sensado de eventos para realizar la clasificación.

8. Trabajo a futuro

Se entiende que las principales lineas a seguir recaen principalmente en un perfeccionamiento del modelo de clasificación, sobre todo juntando más datos, con mejor balance, y que contemplen el medio en el que se insertará el sistema. Si se cuenta con información del Hardware a utilizar, pueden aumentarse las capacidades del modelo. No es descartable probar otros modelos, en particular se adjunta [26, 27] *Few-shot bioacustic event detection*, considerando que el paradigma del *few-shot leraning* ha adquirido cierta popularidad en el último tiempo. El detector de eventos también puede robusteserse en base a datos nuevos, incluso fijando un umbral aprendido de un *dataset* más específico.

9. Conclusión

Luego de un considerable relevamiento de la literatura acerca de la detección de eventos acústicos en general, y el sonido emitido por las cotorras en particular, se concluye que es viable el diseño de un sistema de detección, aplicando modelos que ya son estándares en el área para la clasificación de audio por vía del espectrograma. El desafío principal parece ser continuar robusteciendo el *set* de datos, y la potencial complicación que aún permanece latente es que el nivel de intercambio de información y emisión de sonidos durante la búsqueda de comida o de su ingesta sea suficientemente intensa para la sensibilidad del modelo.

Referencias

[1] SEOBirdLife. Cotorra argentina. https://seo.org/ave/cotorra-argentina/

[2] Tala C, Guzmán P, González S. 2005. Cotorra argentina (Myiopsitta monachus) convidado de piedra en nuestras ciudades y un invasor potencial, aunque real, de sectores agrícolas. Servicio Agrícola y Ganadero – División de Protección de los Recursos Naturales Renovables. Boletín DIPROREN, diciembre 2004 – febrero, 2005. Chile.

[3] Honson A, Avery M, Wright T. 2014. The socioecology of Monk Parakeets: Insights into parrot social complexity.

[4] Smeele S, Senar J, Aplin L, McElreath M. 2023. Evidence for vocal signatures and voice-prints in wild parrot.

[5] Smith-Vidaurre G, Pérez-Marrufo V, Hobson E, Salinas-Melgoza A, Wright T. 2023. Individual identity information persists in learned calls of introduced parrot populations.

[6] Smith-Vidaurre G, Pérez-Marrufo V, Wright T. 2021. Individual vocal signatures show reduced complexity following invasion.

[7] Smith-Vidaurre G, Araya-Salas M, Wright T. 2019. Individual signatures outweigh social group identity in contact calls of a communally nesting parrot

[8] Martella M, Bucher E. 1990. Vocalizations of the Monk Parakeet.

[9] Mott D F. Monk parakeet damage to crops in Uruguay and itr control.

[10] Ministerio de Ganadería, Agricultura y Pesca. Cotorra (Myiopsitta monachus). https://www.gub.uy/ministerio-ganaderia-agricultura-pesca/politicas-y-gestion/cotorra

[11] Viazzi A, Blandón J, Maciel Rios A, Gil González J. 2023. A centered kernel alignment-based strategy for pest evolution tracing: Myiopsitta monachus case.

[12] Prosise J. (2021). Audio classification using convolutional neural networks. https://github.com/jeffprosise/Deep-Learning/blob/master/Audio%20Classification%20(CNN).ipynb

[13] Doshi K. (2021). Audio Deep Learning Made Simple: Sound Classification, Step-by-Step. https://towardsdatascience.com/audio-deep-learning-made-simple-sound-classification-step-by-step-cebc936bbe5

[14] Xiao H, Liu D. 2022. AMResNet: An automatic recognition model of bird sounds in real environment.

[15] Kortas M. (2020). Sound-based bird classification. https://towardsdatascience.com/sound-based-bird-classification-965d0ecacb2b

[16] Mohanty R, Kumar Mallik B, Singh Solanki S. 2020. Automatic bird species recognition system using neural network based on spike.

[17] Kahl S, Wood C, Eibl M, Klinck H. 2021. BirdNET: A deep learning solution for avian diversity monitoring.

[18] Hoffman B, Van Horn G. (2021). From Sound to Images, Part 1: A deep dive on spectrogram creation. https://www.macaulaylibrary.org/2021/07/19/from-sound-to-images-part-1-a-deep-dive-on-spectrogram-creation/

[19] Hoffman B, Van Horn G. (2021). From Sound to Images, Part 2: Spectrogram Image Processing. https://www.macaulaylibrary.org/2021/08/05/from-sound-to-images-part-2-spectrogram-image-processing/

[20] Sawant S, Arvind C, Viral J, Robin V. 2021. Spectrogram cross-correlation can be used to measure the complexity of bird vocalizations.

[21] Deruty E. (2022). Intuitive understanding of MFCCs. https://medium.com/@derutycsl/intuitive-understanding-of-mfccs-836d36a1f779

[22] Rovai M. (2022). TinyML Made Easy: Sound Classification (KWS). https://www.hackster.io/mjrobot/tinyml-made-easy-sound-classification-kws-2fb3ab

[23] Arm. (2021). End-to-end tinyML audio classification with the Raspberry Pi RP2040. https://blog.tensorflow.org/2021/09/TinyML-Audio-for-everyone.html

[24] Maayah M, Abunada A, Al-Janahi K, Ejaz Ahmed M, Qadir J. (2022). LimitAccess: on-device TinyML based robust speech recognition and age classifcation.

[25] Palanisamy K, Singhania D, Yao A. (2020). Rethinking CNN Models for Audio Classification.

[26] I. Nolasco, B. Ghani, S. Singh, E. Vidana-Vila, H. Whitehead, E. Grout, M.G. Emmerson, F. H. Jensen, I. Kiskin, J. Morford, A. Strandburg-Peshkin, L. Gill, H. Pamuła, V. Lostanlen, D. Stowell. (2023). Few-shot bioacoustic event detection at the DCASE 2023 challenge.

[27] Few-shot Bioacoustic Event Detection. https://dcase.community/challenge2022/task-few-shot-bioacoustic-event-detection