



Sistema autónomo ahuyentador de aves utilizando aprendizaje por transferencia

Autonomous bird repellent system using transfer learning

Sistema autônomo de repelência de pássaros usando transfer learning

ARTÍCULO ORIGINAL



Raquel Quelca Cadena 

quelcaraquel1@gmail.com

Universidad Adventista de Bolivia. Cochabamba, Bolivia

Escanea en tu dispositivo móvil
o revisa este artículo en:

<https://doi.org/10.33996/revistaingenieria.v8i21.122>

Artículo recibido 12 de marzo 2024 / Aceptado 3 de abril 2024 / Publicado 15 de mayo 2024

RESUMEN

La producción de mandarina criolla mejorada constituye una fuente económica significativa para numerosos colonizadores en la Colonia Unión Camacho, ubicada en el departamento de La Paz, Bolivia. No obstante, este proceso se ve frecuentemente perturbado por la presencia del ave oropéndola crestada (*Psarocolius decumanus*). Por lo que el objetivo de la actual investigación desarrollar un sistema autónomo de ahuyentamiento de aves basado en la técnica de aprendizaje por transferencia de Machine Learning (ML). La metodología aplicada en el desarrollo de este sistema fue la Extreme Programming (XP). El sistema ahuyentador se sometió a un proceso de entrenamiento y posterior evaluación utilizando conjuntos de datos de entrenamiento y prueba. Los resultados obtenidos demostraron una satisfactoria capacidad de clasificación de los sonidos emitidos por el ave oropéndola crestada, lo que refleja la alta confiabilidad del sistema propuesto. Por lo que, destaca la capacidad para aprovechar las tecnologías de vanguardia, en particular la Inteligencia Artificial, y aplicarlas al sector agrícola. De esta manera, se contribuye al desarrollo económico y educativo de la comunidad.

Palabras clave: Aprendizaje por transferencia; Aplicación móvil; Agricultura; Inteligencia artificial

ABSTRACT

The production of improved Creole mandarin constitutes a significant economic source for numerous colonizers in the Colonia Unión Camacho, located in the department of La Paz, Bolivia. However, this process is frequently disturbed by the presence of the crested oriole bird (*Psarocolius decumanus*). Therefore, the objective of the current research is to develop an autonomous bird scaring system based on the Machine Learning (ML) transfer learning technique. The methodology applied in the development of this system was Extreme Programming (XP). The deterrent system underwent a training process and subsequent evaluation using training and test data sets. The results obtained demonstrated a satisfactory classification capacity of the sounds emitted by the crested oriole bird, which reflects the high reliability of the proposed system. Therefore, the ability to take advantage of cutting-edge technologies, particularly Artificial Intelligence, and apply them to the agricultural sector stands out. In this way, it contributes to the economic and educational development of the community.

Key words: Transfer learning; Mobile application; Agriculture; Artificial intelligence

RESUMO

A produção de mandarina criolla melhorada constitui uma fonte econômica significativa para numerosos colonizadores da Colônia União Camacho, localizada no departamento de La Paz, Bolívia. No entanto, este processo é frequentemente perturbado pela presença do papa-figo-de-crista (*Psarocolius decumanus*). Portanto, o objetivo da presente pesquisa é desenvolver um sistema autônomo de espantar pássaros baseado na técnica de aprendizagem por transferência de aprendizado de máquina (ML). O sistema dissuasor passou por um processo de treinamento e posterior avaliação utilizando conjuntos de dados de treinamento e teste. Os resultados obtidos demonstraram uma capacidade satisfatória de classificação dos sons emitidos pelo papa-figo, o que reflete a alta confiabilidade do sistema proposto. Destaca-se, portanto, a capacidade de aproveitar tecnologias de ponta, nomeadamente a Inteligência Artificial, e aplicá-las ao setor agrícola. Desta forma, contribui para o desenvolvimento econômico e educacional da comunidade.

Palavras-chave: Aprendizagem por transferência; Aplicativos móveis; Agricultura; Inteligência artificial

INTRODUCCIÓN

En la actualidad, la tecnología desempeña un papel fundamental en la simplificación y optimización de diversas actividades, reduciendo tiempos, costos y esfuerzos en su ejecución. El sector agrícola no es una excepción a esta tendencia. Las nuevas tecnologías se caracterizan por ser cada vez más eficientes, económicas

y respetuosas con el medio ambiente, lo que posibilita la automatización de tareas que suelen ser agotadoras, complejas e incluso riesgosas para los agricultores. Este fenómeno es evidente en el contexto de los productores de mandarina criolla mejorada, quienes se ven afectados por la presencia y acciones del ave oropéndola (Scotta, et al., 2018), tal como se ilustra en la Figura 1.



Figura 1. Daños ocasionados por el ave oropéndola.

Los cítricos, propiamente arbóreos y explotados de manera intensiva, también han mostrado ciertas bondades para la conservación de las aves residentes y migratorias. A pesar de la simplicidad de su estructura botánica, son capaces de atraer un elevado número de aves a su interior desde formaciones boscosas aledañas, principalmente para su abastecimiento alimentario. En este sentido en estudio de Montes y Solórzano, (2024), en Venezuela en el cultivo de mandarina se capturaron 50 especies en 200 capturas, por lo que su riqueza resultó moderada. Sin embargo, su número resultó inferior al de otro

cítrico (naranja) previamente estudiado, pero superior al de otros cultivos como el durazno, aguacate y banano. De ellas, la Reinita *Coereba flaveola* resultó la más abundante con el 13,0% de las capturas totales, seguida de la Tángara Monjita Tangara cayana (8%), el Azulejo de Jardín *Thraupis episcopus* (7,5%), el Atrapamoscas Color Ratón *Phaeomyias murina* (6,0%), el Chocolatero *Tachyphonus rufus* (4,5%), el Diamante Gargantiverde *Amazilia fimbriata* y la Paloma Turca *Leptotila verreauxi* (4%).

Con el propósito de contrarrestar esta problemática, se han empleado diversos métodos

para ahuyentar a las aves. Entre las prácticas más habituales se encuentran el uso de espantapájaros y la colocación de cintas reflectivas o discos alrededor de las plantas. Sin embargo, cabe destacar que los discos solo brindan protección en condiciones de sol y viento, debido a su falta de movilidad. Adicionalmente, se han recurrido a resorteras y sustancias químicas como medidas de control, pero estas acciones han resultado perjudiciales para la fauna de Bolivia. Otra estrategia empleada es la presencia física en el lugar de las plantaciones y la generación de ruido, aunque este enfoque no puede mantenerse de manera constante debido a las responsabilidades adicionales que enfrentan los agricultores (Sheriff, 2015).

Por lo tanto, con el objetivo de mejorar la eficacia de los métodos mencionados previamente, se propuso el desarrollar un sistema autónomo de ahuyentamiento de aves basado en la técnica de aprendizaje por transferencia de Machine Learning (ML). Para este propósito, se empleó YAMNet como modelo pre-entrenado. YAMNet es un clasificador de eventos de audio que opera tomando la forma de onda de audio como entrada y efectúa predicciones independientes para cada uno de los 521 eventos de audio. La implementación de esta tecnología se plantea como una solución prometedora para mitigar el impacto de las aves en las plantaciones de mandarina criolla mejorada.

MÉTODO

La metodología de desarrollo seleccionada para este proyecto es la Programación Extrema (Extreme Programming, XP). La Programación Extrema se basa en la aplicación rigurosa de las mejores prácticas de ingeniería de software y fue concebida por Kent Beck y Ward Cunningham. El nombre de esta metodología alude a su enfoque extremo en la aplicación de estas buenas prácticas (Letelier y Penadés, 2012). El proceso de desarrollo del proyecto se divide en las siguientes fases:

Fase de Exploración. En esta etapa inicial, se definen las historias de usuario, se realizan pruebas de tecnologías y se exploran las posibilidades de la arquitectura del sistema.

Fase de Planificación. En esta fase se determina el orden de implementación de las historias de usuario, se establecen los tiempos estimados de desarrollo para cada historia y se asigna una prioridad de implementación a cada una de ellas.

Fase de Iteraciones y Producción. En esta etapa, se lleva a cabo la construcción del software en sí. Cada iteración se enfoca en la implementación de un conjunto específico de características del sistema. La Tabla 1 proporciona una estimación del esfuerzo y un plan de entrega para las historias de usuario durante las iteraciones.

Este enfoque de desarrollo estructurado garantiza la entrega gradual y continua de funcionalidades del sistema, permitiendo una adaptación ágil a los cambios y requisitos emergentes a lo largo del proceso.

El sistema propuesto se compone de dos partes fundamentales: los componentes de hardware y los componentes de software.

Componentes de Hardware

Los componentes de hardware esenciales para el funcionamiento del sistema son los siguientes:

Dispositivo Móvil. Este dispositivo actúa como el componente principal de hardware. Se utiliza para instalar la aplicación que ejecutará el modelo y aprovecha su micrófono incorporado para capturar los sonidos ambientales.

Cable OTG (On-The-Go). El cable OTG permite la conexión de dispositivos externos mediante USB al dispositivo móvil. En este contexto, se utiliza para conectar el dispositivo móvil al dispositivo Arduino y facilitar la transferencia de datos.

Arduino. Se emplea una placa electrónica de hardware libre con un microcontrolador reprogramable y una serie de pines que posibilitan las conexiones con diferentes sensores. En el sistema, el dispositivo Arduino recibe datos y, mediante un programa específico, controla la activación de la bocina.

Relé. funciona como un interruptor eléctrico que permite o corta el paso de la corriente eléctrica cuando se activa o desactiva, respectivamente. Este interruptor se acciona eléctricamente, de modo que cuando el dispositivo Arduino envía una señal activa, permite el flujo de corriente eléctrica para encender la bocina.

Bocina. Para el sistema se utiliza una bocina tipo claxon automotriz que normalmente se emplea para alertar a otros conductores o usuarios de la carretera. En el contexto del sistema propuesto, la bocina cumple la función de ahuyentar a las aves mediante el sonido que emite.

Batería. Se emplea una batería automotriz de 12V como fuente de energía para alimentar la bocina.

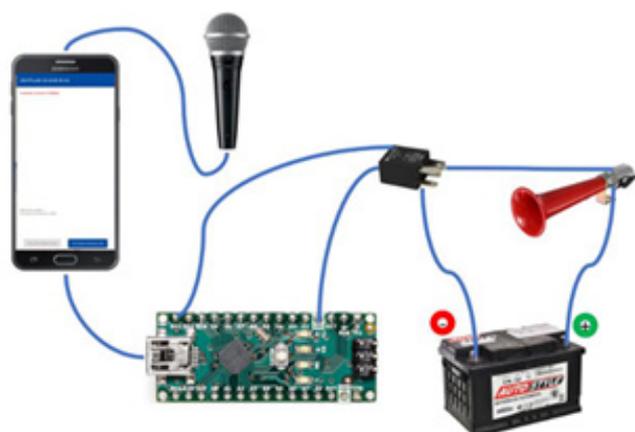


Figura 2. Componentes de hardware del sistema.

Componentes de Software

El sistema propuesto también se basa en varios componentes de software, siendo los principales los siguientes:

Python. es un lenguaje de programación interpretado, orientado a objetos y de alto nivel con semántica dinámica. Es ampliamente conocido por su capacidad para el desarrollo de modelos de aprendizaje automático (Machine Learning, ML) debido a su amplio conjunto de paquetes que simplifican la codificación. Python es de código abierto y cuenta con una abundante documentación de alta calidad y recursos (Rozo, et al., 2020). En este proyecto, se utiliza Python para escribir el código del modelo de aprendizaje en todas sus etapas.

YAMNet. es una red neuronal profunda diseñada para la clasificación de eventos de audio y realiza predicciones independientes para cada uno de los 521 eventos del conjunto de datos de AudioSet (Brusa, et al., 2021). YAMNet se emplea en la creación de un clasificador de sonido personalizado para detectar la presencia de aves en el entorno. En la Figura 3 se puede observar el proceso de construcción de un modelo de aprendizaje.

TensorFlow. es una librería de código abierto desarrollada por Google que se utiliza para la construcción y entrenamiento de redes neuronales. Dispone de un ecosistema completo de herramientas, bibliotecas y recursos de la comunidad que fomentan la innovación en el campo del aprendizaje automático (ML) (Valle-Barrio, 2018).

Matplotlib. permite la creación de gráficos en 2D y se integra estrechamente con la biblioteca NumPy y otras. Además, ofrece funcionalidades interactivas como zoom, desplazamiento y la capacidad de guardar figuras en varios formatos (Agundis-Martínez, 2023).

NumPy. es una biblioteca que proporciona un objeto de matriz multidimensional y facilita una amplia gama de operaciones rápidas con matrices. Estas operaciones incluyen matemáticas, lógica, manipulaciones de formas, clasificación, selección, transformadas discretas de Fourier, álgebra lineal básica, estadísticas y simulación aleatoria, entre otras (Bauer y Garland, 2019).

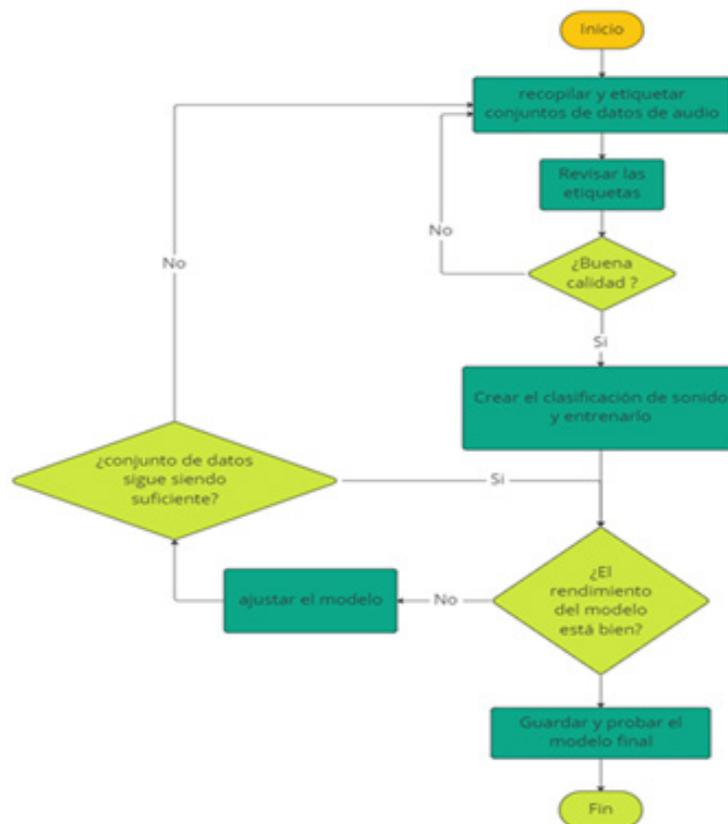


Figura 3. Proceso de construcción de un clasificador de sonido.

RESULTADOS Y DISCUSIÓN

El desarrollo del software propuesto se llevó a cabo en múltiples iteraciones, cada una enfocada en aspectos específicos del sistema. A continuación, se describe cada iteración junto con sus resultados correspondientes:

Iteración 1: Diseño de la interfaz gráfica y módulo inicial

En esta primera iteración, se diseñó la interfaz gráfica del sistema ahuyentador del ave oropéndola crestada. La interfaz incluye

dos iconos principales, uno relacionado con el modelo YAMNet y otro destinado a la clasificación del ave oropéndola. Posteriormente, se implementó la identificación del ave mediante una alerta utilizando un modelo de prueba de TensorFlow, ya que aún no se contaba con un modelo personalizado. Además, se estableció la comunicación con la bocina a través del cable OTG y un dispositivo Arduino, y se recolectaron audios del ave oropéndola crestada y otras aves de la región, Figura 4.

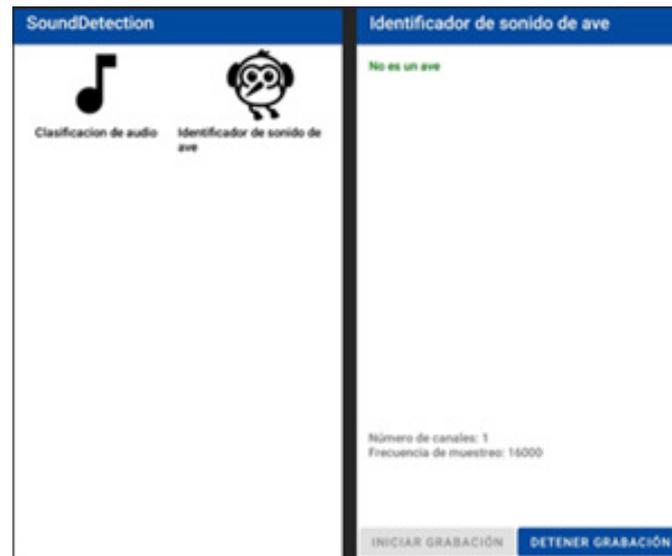


Figura 4. Diseño de la interfaz gráfica.

La realización de este proyecto planteó diversos desafíos y obstáculos que requirieron soluciones creativas y enfoques cuidadosos. Uno de los desafíos más significativos fue la adquisición de un conjunto de datos adecuado para el entrenamiento del modelo de aprendizaje automático. Se optó por utilizar la fuente xeno-canto.org para recopilar audios relacionados con la oropéndola crestada (*Psarocolius decumanus*), identificada mediante su nombre científico. Este proceso de recolección implicó la escucha individual de cada audio, ya que se encontraron casos en los que el canto del ave se presentaba en una parte del audio, mientras que el resto contenía silencio o ruido. Esta meticulosa recopilación de datos fue esencial para garantizar la calidad y utilidad del conjunto de datos.

El desarrollo continuo en el área de la bioacústica sigue siendo hoy promisorio en teoría (Rempel, et al., 2005). Aunque existe

más de una década de investigaciones en bioacústica, existen diversos enfoques para llevar a cabo la identificación debido a que la literatura concerniente se encuentra dispersa entre biología e ingeniería. A pesar de que existen avances tecnológicos para la aplicación en la conservación de especies, algunas de ellas resultan muy costosas tanto en espacio y tiempo (como el caso de la telemetría la cual es utilizada para el seguimiento de aves, seguimiento de rutas migratorias, resguardo de anidación, entre otros) (Pedrosa, et al., 2018).

Además, era fundamental disponer de audios de aves que no pertenecieran a la clase de la *Oropéndola crestada*. Para esto, se agruparon varias especies de aves de la región que no representaban una amenaza para los cultivos. Sin embargo, la obtención de grabaciones de estas aves no siempre fue sencilla, ya que sus nombres locales no estaban disponibles en las fuentes

Quelca R.

en línea. Este desafío destacó la importancia de la colaboración con la comunidad local y el conocimiento tradicional en la identificación de aves.

En este sentido Contreras y Ortiz, (2023) al utilizar el desarrollo de un sistema Operativo Raspberry Pi Os (Raspbian) para el ahuyentar aves utilizaron este sistema operativo debido a que ofrece una combinación de compatibilidad con otros sistemas operativos, facilidad de uso ya que tiene una interfaz de usuario amigable y familiar, amplia gama de aplicaciones, soporte de la comunidad activa de usuarios y desarrolladores que brindan soporte y comparten recursos en línea como por ejemplo una amplia variedad de tutoriales, documentación y proyectos relacionados con el Raspberry Pi OS, además de actualizaciones regulares, lo que lo convirtió en

el más adecuado. Se emplea el Raspberry Pi Os (Legacy) con Debian Buster de 32 bits y entorno de escritorio Kernel de Linux 5.10.x

Iteración 2: Creación del conjunto de datos y modelo de aprendizaje

En esta iteración, se procedió a crear el conjunto de datos que serviría para entrenar el modelo de aprendizaje automático (ML) utilizando la técnica de aprendizaje por transferencia. El conjunto de datos incluye metadatos en un archivo CSV y una estructura organizada en carpetas para las clases de aves. Los archivos de audio siguen especificaciones detalladas. Posteriormente, se utilizó Google Colab para crear y entrenar el modelo de aprendizaje automático, Figura 5.

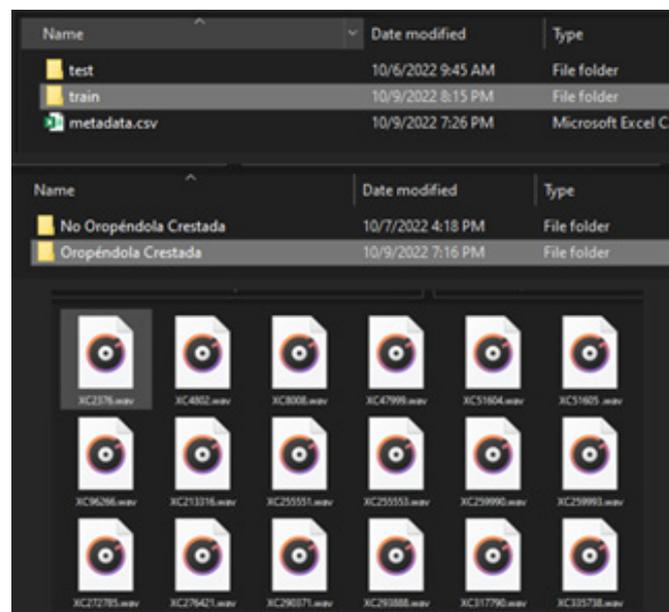


Figura 5. Diseño de la interfaz gráfica.

La creación del conjunto de datos personalizado siguió especificaciones rigurosas en cuanto a la tasa de muestreo, el formato de audio y otros detalles técnicos para garantizar la compatibilidad con el modelo de aprendizaje automático. Posteriormente, se empleó Google Colab para llevar a cabo el entrenamiento del modelo. Durante este proceso, se ajustaron parámetros y se utilizaron configuraciones predeterminadas para entrenar el modelo.

Iteración 3: Entrenamiento del modelo y evaluación

En esta tercera iteración, se obtuvo el modelo entrenado y se realizó una

evaluación detallada del mismo. Los datos se comprimieron y almacenaron en un repositorio público de GitLab, permitiendo su descarga y descompresión. Se entrenó el modelo de aprendizaje automático utilizando TensorFlow y se ajustaron parámetros necesarios. Se midió el desempeño del modelo mediante una matriz de confusión y se calcularon métricas como precisión, sensibilidad y puntuación F1. El modelo demostró una precisión del 84%, una sensibilidad del 91% y una puntuación F1 del 87%, indicando un buen rendimiento.

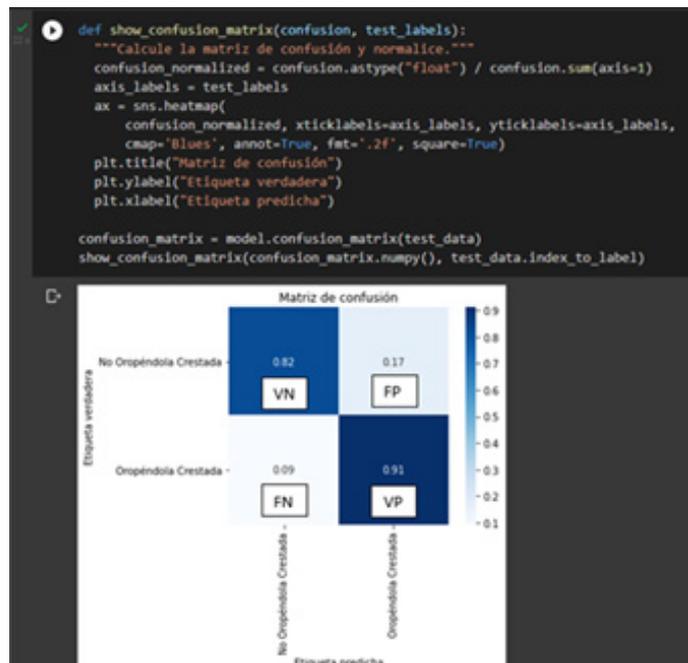


Figura 6. Matriz de confusión.

La evaluación del desempeño del modelo se realizó mediante una matriz de confusión y el cálculo de métricas clave, como la precisión, la sensibilidad y la puntuación F1. Los resultados obtenidos demostraron que el modelo tenía una precisión del 84%, una sensibilidad del 91% y una puntuación F1 del 87%. Estas métricas indican un buen rendimiento del modelo, lo que es esencial para su utilidad en aplicaciones del mundo real.

Al emplear redes neuronales Contreras y Ortiz, (2023), para la creación del Dataset se utilizó el método de la gaussiana adaptativa este consiste en la sustracción de fondo y la clasificación de objetos. Los objetos en movimiento se detectan mediante la sustracción de fondo y solo el ave se extrae mediante la clasificación de objetos.

Iteración 4: Integración de la bocina

En esta cuarta iteración, se integró la bocina en el sistema ahuyentador de aves, como se muestra en la Figura 7. Esta fase representa la culminación de las fases de iteración y producción

del proyecto. Se logró una integración exitosa de todos los componentes, cumpliendo con los requerimientos de los usuarios en cada una de las iteraciones anteriores.

Finalmente, se logró la integración exitosa de la bocina en el sistema ahuyentador de aves, culminando así el proyecto. El sistema completo satisface los requerimientos de los usuarios en todas las iteraciones del desarrollo y representa una solución efectiva para mitigar los problemas causados por el ave oropéndola crestada en las plantaciones de mandarina criolla mejorada. En resumen, a lo largo de las iteraciones del proyecto, se diseñó la interfaz gráfica, se creó un conjunto de datos y un modelo de aprendizaje, se entrenó y evaluó el modelo, y finalmente se integró la bocina en el sistema ahuyentador de aves. Cada fase se enfocó en alcanzar objetivos específicos y, en general, se cumplieron satisfactoriamente los requerimientos de los usuarios, logrando mejoras sustanciales en el sistema.



Figura 7. Pruebas del sistema.

CONCLUSIONES

La atracción por playback se puede aplicar en la marcación de individuos y para una captura selectiva (dado que la grabación es especie-específica). De esta forma las aves podrían ser atraídas hacia sitios seguros y por ende ayudar en la conservación de especies las cuales son atraídas por los parlantes. En cuanto al ahuyentamiento, este método es utilizado cuando se requiere evitar en cierta área la presencia de aves. Este es el sustituto de técnicas que implicaban barreras o métodos pirotécnicos. En su lugar, se producen sonidos naturales o de gran intensidad capaces de espantar a las aves y evitar así pérdidas económicas o posibles accidentes en áreas donde su presencia sería perjudicial. Para este último caso, es necesario tomar en cuenta la posible extinción de respuesta asociada al estímulo; por lo que será necesario cambiar de playback con frecuencia, aunque se recomienda la combinación de varios métodos para asegurar la efectividad del espantamiento (Pedroza, et al., 2016).

Este proyecto ilustra cómo la combinación de tecnología de vanguardia, colaboración con la comunidad local y un enfoque metódico puede conducir al desarrollo exitoso de soluciones innovadoras en el campo de la agricultura. El enfoque en la calidad de los datos, el entrenamiento del modelo y la evaluación meticulosa son factores clave para el logro de un sistema eficaz de ahuyentamiento de aves que beneficie tanto a los agricultores como al medio ambiente.

Se ha logrado la construcción exitosa del modelo de aprendizaje utilizando YAMNet. Este proceso se ha llevado a cabo haciendo uso de las funcionalidades proporcionadas por Model Maker de TensorFlow.

El modelo ha sido entrenado empleando un conjunto de datos que ha demostrado buenos resultados, alcanzando una precisión del 87%. Para medir la precisión y calidad del modelo, se ha utilizado la herramienta de matriz de confusión. Posteriormente, se ha exportado el modelo a formato tflite para su implementación en la aplicación Android.

En cuanto al aspecto hardware y la interfaz gráfica, inicialmente se consideró desarrollar el proyecto en PyQt y ejecutarlo en un dispositivo Raspberry Pi. Sin embargo, se han experimentado dificultades en la instalación de la librería TensorFlow en esta plataforma, lo que llevó a la decisión de desarrollar el proyecto en un dispositivo Android. Para la comunicación entre la aplicación y el hardware se ha empleado un cable serial OTG, un Arduino y un relé para el control de la activación de la bocina.

CONFLICTO DE INTERESES. Los autores declaran que no existe conflicto de intereses para la publicación del presente artículo científico.

REFERENCIAS BIBLIOGRÁFICAS

- Agundis-Martínez, A. F. (2023). Evaluación de algoritmo de Aprendizaje Automático mediante Bosques Aleatorios para el seguimiento de variables críticas en un proceso de manufactura. <http://51.143.95.221/handle/TecNM/6959>
- Bauer, M., y Garland, M. (2019). Legate NumPy: Accelerated and distributed array computing. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. 1 (23),1-23. <https://doi.org/10.1145/3295500.3356175>
- Brusa, E., Delprete, C., y Di Maggio, L. G. (2021). Deep transfer learning for machine diagnosis: From sound and music recognition to bearing fault detection. *Applied Sciences*, 11(24), 11663. <https://doi.org/10.3390/app112411663>
- Contreras, I., y Ortíz, F. (2023). Sistema automatizado ahuyentador de aves para las plantaciones de arroz. Universidad Agraria del Ecuador. <https://cia.uagraria.edu.ec/Archivos/ORTIZ%20ROBLES%20ALEXANDER%20FRANCISCO.pdf>
- Letelier, P., y Penadés, M. C. (2012). Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP). Universidad Politécnica de Valencia. https://www.researchgate.net/profile/Patricio-Letelier/publication/28109707_Metodologias_agiles_para_el_desarrollo_de_software_eXtreme_Programming_XP/links/54ad00f10cf2479c2ee86820/Metodologias-agiles-para-el-desarrollo-de-software-eXtreme-Programming-XP.pdf
- Montes, N., y Solórzano, A. La comunidad de aves en un cultivo de mandarinas del norte de Venezuela/Bird Community in a Tangerine Field from Northern Venezuela. *Revista Venezolana de Ornitología*, 2(1), 4-15. <https://digitalcommons.usf.edu/cgi/viewcontent.cgi?article=1000&context=rvo>
- Pedrosa, Á. D., De la Rosa Vargas, J. I., y Valdez, R. R. (2018). Escuchando a la naturaleza: Del reconocimiento de voz a la bioacústica. *Pistas Educativas*, 38(120). file:///C:/Users/Danis/Downloads/598-1749-2-PB.pdf
- Rempel, R. S., Hobson, K. A., Holborn, G., Van Wilgenburg, S. L., y Elliott, J. (2005). Bioacoustic monitoring of forest songbirds: interpreter variability and effects of configuration and digital processing methods in the laboratory. *Journal of Field Ornithology*, 76(1), 1-11. <https://doi.org/10.1648/0273-8570-76.1.1>
- Rozo, C. E., Cusba, J. E., Medina, L. F., y León, C. J. (2020). Herramientas de analítica para la exploración de datos, 1ra ed., Colombia: Creative Commons, 22-57. https://herramientas.datos.gov.co/sites/default/files/2020-11/Inventario%20herramientas%20anal%C3%ADtica_0.pdf
- Scotta, R. R., Canavelli, S. B., y Lutz, A. L. (2018). Percepción del daño causado por aves en frutales y alternativas de manejo en el centro norte santafesino. *Fave. Sección ciencias agrarias*, 17(1), 45-55. <http://www.scielo.org.ar/pdf/fave/v17n1/v17n1a04.pdf>
- Sheriff, R. (2015). Comparación de dos parcelas con cítricos en sistemas agroforestales con sucesiones vegetales en la comunidad de San Pedro del Zapallar, Monteagudo: Comparison between two citrus plots in successional agroforestry systems in San Pedro del Zapallar, Monteagudo, Bolivia. *AGRO-ECOLÓGICA*, 2(1), 246-256. <https://revistas.usfx.bo/index.php/rae/article/view/108/92>
- Valle-Barrio, A. (2018). Aplicación de Tensorflow en deep learning. Universidad Politécnica de Madrid, Escuela Técnica Superior de Ingenieros de Telecomunicación. https://oa.upm.es/53815/1/TESIS_MASTER_ALFREDO_VALLE_BARRIO_2018.pdf