

Monitoreo de frecuencias de vuelo y pausas en vuelo estacionario de Colibríes a través de técnicas de visión por computador.

RESUMEN:

Los colibríes exhiben características y comportamientos únicos, incluyendo roles cruciales como polinizadores y una extraordinaria habilidad aerodinámica. Son capaces de volar en todas las direcciones, incluyendo el vuelo estacionario, un rasgo más común en los insectos que en otras aves. Esta patente propone un sistema para explorar aspectos del vuelo, como la estimación de la frecuencia de aleteo y un comportamiento único: las pausas realizadas durante el vuelo.

La estimación de las frecuencias de aleteo es un aspecto clave en el estudio del vuelo de las aves, ya que permite determinar elementos energéticos y patrones de comportamiento, entre otras características. Los colibríes son sujetos ideales para probar métodos de estimación debido al rápido movimiento de sus alas y su particular aerodinámica, resultado de su diversificación ecológica, adaptación a ambientes de altura y la selección natural.

Además, durante el vuelo estacionario, en algunas ocasiones y de manera momentánea, los colibríes mantienen sus alas abiertas por fracciones de segundo, generando lo que denominamos "pausas en vuelo". Estas pausas podrían proporcionar nuevos conocimientos sobre el comportamiento de las aves, así como sobre su comunicación e interacción sensorial.

Esta patente presenta un método para la detección, clasificación, seguimiento y monitoreo de las frecuencias de aleteo y las pausas en vuelo de estas aves. Los análisis se realizan en videos de alta velocidad para estimar de manera precisa los patrones deseados, utilizando técnicas de visión por computadora y análisis de señal. La automatización del procesamiento de video facilita la estimación de características del vuelo, reduciendo el tiempo de procesamiento de los datos recolectados y, en consecuencia, contribuyendo a un entendimiento más profundo de las dinámicas de vuelo del colibrí, su comportamiento y las señales de comunicación e interacción con su entorno.

SECTOR DE LA TÉCNICA:

Hace referencia al monitoreo y estudio de comportamiento animal en biología utilizando técnicas de inteligencia artificial específicamente en la visión por computador.

CLASIFICACIÓN:

G06F 18/00 Pattern recognition

G06F 18/40 Software arrangements specially adapted for pattern recognition, e.g. user interfaces or toolboxes therefore.

ESTADO DE LA TÉCNICA:

La conservación de los colibríes tiene un papel relevante, ya que permite proteger no únicamente las especies, sino también ecosistemas completos debido a su papel de polinizador. La relación mutualista existente entre animales nectarívoro (modo de alimentación inicialmente de insectos y más adelante adoptado por algunas aves) y plantas florales ha permitido una evolución convergente en rasgos y diversidad [1]; en las aves, por ejemplo, existe una coincidencia entre la evolución generada por la relación entre el tamaño y la forma tanto de picos como de flores [1]. En el departamento de Nariño en Colombia, se encuentra una de las comunidades de colibríes más diversas del mundo [2]. En la reserva Río Ñambí, existen, por lo menos, 29 especies de colibríes de la familia Trochilidae de las cuáles 9 son endémicas en Colombia [2]. Los colibríes (familia Trochilidae) son aves reconocidas por su vuelo ágil y habilidad única entre todas las aves para sostener un vuelo estacionario [3, 4]. Esta habilidad es garantizada por una combinación entre la adaptación anatómica y fisiológica (por ejemplo, la adaptación de los huesos de las alas, entre otros, o el aumento de la capacidad aeróbica y frecuencias altas en el aleteo) y cuyo estudio continúan siendo de gran interés [3]. Este tipo de vuelo, permite realizar un movimiento aerodinámico único en aves a cambio de un alto costo de consumo de energía [5]. El estudio de la frecuencia de vuelo de estas aves tiene implicaciones en diferentes campos, por ejemplo, en la robótica, la ingeniería aeroespacial y la biología evolucionaria (e.g., Altshuler et al. 2004 [6], Tobalske et al. 2007 [7], Nan et al. 2019 [8], Tu et al. 2020 [9], Wilcox and Clark 2022 [10], Zhao et al. 2022 [11], Jensen et al. 2024 [12]).

La versatilidad de su vuelo permite el desarrollo de maniobras como el vuelo estacionario, el vuelo en reversa y los rápidos giros (e.g., Martínez et al. 2015 [13], Dakin et al. 2018 [14]). Estas capacidades son cruciales para su supervivencia y alimentación ya que les permite acceder al néctar de las flores con precisión y eficiencia (e.g., Stiles 1971 [15], Inouye et al. 1991 [16]). Sin embargo, la frecuencia de aleteo de la mayoría de las especies permanece desconocida, en su mayor parte por limitaciones logísticas (Por ejemplo: capturar y procesar vídeos en alta velocidad). Los colibríes muestran variaciones mínimas en la frecuencia de aleteo en las diferentes condiciones de vuelo, incluyendo el vuelo hacia adelante y el vuelo estacionario (Chai et al., 1999 [17]; Tobalske et al., 2007 [7]; Clark and Mistick, 2018

[18]), incluso cuando se enfrenta a retos aerodinámicos (e.g., Chai 1997 [19], Díaz-Salazar et al. 2023 [20]), por lo tanto, la caracterización de la frecuencia de aleteo sería informativa en diferentes contextos.

Greenwalt (1960) [21] remarcó que alométricamente, los colibríes tienen una menor frecuencia de aleteo que la que sería esperada para su tamaño. Por ejemplo, un pájaro carbonero de 10 gramos tiene una frecuencia de aleteo mayor que un colibrí del mismo peso. Wilcox y Clark (2022) [10], derivaron la relación alométrica esperada para la frecuencia de aleteo del colibrí, teniendo en cuenta la longitud alométrica del ala. Comparado a la mayoría de las aves, los colibríes tienen alas más largas relativas al tamaño de su cuerpo. Esta longitud alométrica del ala, les permite tener una alta capacidad para la maniobrabilidad y vuelo sostenido. Los descubrimientos de Wilcox y Clark [10] revelan que la pendiente alométrica de la frecuencia de vuelo en 30 especies de 37 del colibrí abeja, coinciden con los valores predichos. Resaltando que la estructura única del ala y la frecuencia de aleteo tienen un papel fundamental en las mecánicas de vuelo y presentan un área significativa de investigación futura (Greenewalt, 1960 [21]; Wilcox and Clark, 2022 [10]). Sin embargo, para lograrlo se necesita recolectar datos en gran proporción de más de 350 especies de colibríes, a través de todos los rangos de altura y latitud en la que habitan.

Actualmente, las estimaciones de frecuencia de aleteo se han desarrollado de forma visual, digitalizando cada uno de los cuadros de video y contando cada uno de los aleteos en videos de alta velocidad o procesando las grabaciones de sonido de vuelo estacionario (e.g., Steen 2014 [22]; Sotavalta 1952 [23]). Automatizar el análisis de video podría ayudar en las etapas de preprocesamiento, aliviando la necesidad de utilizar una cantidad significativa de tiempo y esfuerzo. Sin embargo, existen algunas limitaciones en automatizar los datos a partir de una colección de videos. En primer lugar, se necesita una gran cantidad de videos para entrenar un modelo de aprendizaje de máquina. En segundo lugar, la iluminación, la velocidad de fotogramas, la nitidez, la orientación del colibrí, la oclusión de otros objetos, entre otros, dificultan la tarea para determinar los patrones de movimiento y la calidad de los datos. En tercer lugar, en el aprendizaje de aprendizaje de máquina supervisado estos datasets deben ser anotados por un experto humano (e.g., etiquetar manualmente las cajas que delimitan el objeto de estudio en los frames de video). Estas anotaciones son típicamente realizadas cuadro por cuadro y pueden tomar una gran cantidad de tiempo y esfuerzo.

En este contexto, métodos recientes de aprendizaje zero-shot, son enfoques de machine learning donde el modelo es capaz de analizar data que no ha sido visto anteriormente por el mismo (Xian et al., 2019 [24]). Estos métodos utilizan modelos pre-entrenados (Disponibles online, e.g., Yolo models [25], Open Clip Model 2024 [26]) que han sido entrenados en diferentes tareas (e.g., detección de objetos o clasificación), utilizando datasets de referencia que pueden ser adaptados a nuevas

y desconocidas tareas. Este proceso podría funcionar para una pregunta de investigación particular, incluso cuando los modelos no fueron entrenados específicamente para el problema dado. Aún así es posible que generen resultados adecuados. Nguyen et al. 2024 [27], usa esta técnica para detectar y segmentar píxeles donde los animales están presentes en la imagen, como delfines, sapos y aves, entre otros, con la particularidad que la textura ambiente del fondo y los colores tienden a mezclarse con los propios de los animales; en su estudio se utiliza un extractor de características de deep learning y se comparan por medio de la similaridad del coseno; En términos de las cajas de delimitación utilizan el detector YOLO (You Only Look Once) — ampliamente reconocido en el campo de la visión por computador (see Jiang et al. 2022 [28]) — y ha sido utilizado por diferentes investigadores (e.g., Ma and Yang 2022 [29]).

La versatilidad del algoritmo de CLIP (Radford et al., 2021 [30]) ha permitido mejorar los componentes de algoritmos de zero-shot. Shin et al 2023 [31], utilizaron este método para desarrollar un sistema de segmentación. También se ha utilizado en la recategorización de aves a partir de la descripción de las características de color y la textura de las alas (Wu and Maji, 2022 [32]). Los enfoques de visión por computador se han utilizado también para la estimación de frecuencia de aleteo, por ejemplo en la segmentación y seguimiento en videos infrarrojos de murciélagos, para evaluar la posición de las alas relativa a su cuerpo en estado de vuelo (Breslav et al., 2012 [33]). También, Ling et al. 2018 [34], utilizan el análisis tridimensional para capturar imágenes de video estero de cuervos para seguir las velocidades de movimiento, la aceleración y la frecuencia de aleteo, asegurando que estos cálculos ayudan a entender la cinemática de vuelo de las aves.

Para los colibríes, los estudios en visión por computador se han centrado en la clasificación de cuadros de video identificando colibríes en sus nidos (Serrano et al., 2018 [35]); o en detectores como el propuesto por Weinstein [36, 37], quien compara cada fotograma de video de una imagen con el fondo de la imagen para determinar la presencia de un nuevo objeto para filtrar videos (una alternativa a la cámara trampa y sistema de captura propuesto por Rico-Guevara and Mickley 2017 [38]). Martinez y colaboradores 2015 [13], usan el flujo óptico denso para caracterizar el movimiento de las alas de los colibríes, midiendo la aceleración global angular y la deformación global de las alas. El vuelo de estas aves ha inspirado estudios en el desarrollo de vehículos de aire micro (Micro Air Vehicles MAVs e.g., Nan et al. 2019 [8], Tu et al. 2020 [9]). Por último, Zhao y colaboradores (Zhao et al., 2022 [11]) diseñaron un algoritmo de optimización para procesos de ingeniería a través de la simulación del vuelo del colibrí.

Dada el interés multidisciplinario de este tema, y la ausencia de estudios enfocados en el desarrollo de métodos de estudio de las características de vuelo de forma eficiente a través de especies, nos enfocamos en el desarrollo de un sistema automático para la detección, clasificación, seguimiento y cuantificación de

frecuencias de aleteo y pausas en vuelo de colibríes. Con esta patente buscamos la generación de herramientas para ampliar el conocimiento existente y que fomenten la conservación de la especie así como analíticas aéreas particulares; permitiendo realizar un análisis visual focalizado a partir de video de estos animales extraordinarios.

DESCRIPCIÓN BREVE DE LA INVENCION:

Este método realiza una detección y seguimiento automático de colibríes para la medición de frecuencia de vuelo y pausas en vuelo utilizando videos de forma offline. Un diagrama de bloques del sistema completo se aprecia en la FIG 1; El método planteado provee como resultado un video con la detección realizada, un archivo con la información de la caja de detección, clasificación de objeto al interior de la caja y número de seguimiento del (los) objeto(s) localizados en cada frame; adicionalmente, se genera un gráfico con la descripción de comportamiento del (los) colibrí(es) incluyendo la detección de pausas en vuelo (detención de aleteo) suspendidas, un ejemplo de dichas pausas se presenta en la FIG 2. El sistema incluye: (1.1) un módulo de obtención de vídeo y separación de frames; (1.2) un módulo de detección de objetos separando objetos dinámicos de la imagen de el fondo; (1.3) un método de clasificación que permite la localización tanto de los colibríes así como otros objetos presentes en los videos y especificados a partir de una entrada de texto por el usuario (por ejemplo: flores, alimentadores, personas, árboles, entre otros); (1.4) Un sistema de seguimiento a partir de la comparación de un vector de características del frame anterior con el frame siguiente realizando el seguimiento de los objetos detectados a través del video; y finalmente, (1.5) un módulo de monitoreo de frecuencias de vuelo y pausas aéreas, a través medición de movimiento entre las cajas de detección a través de los frames con el fin de determinar las frecuencia de vuelo y pausas que pueden realizar los colibríes mientras están suspendidos en el aire.

DESCRIPCIONES DE LAS FIGURAS:

FIG. 1 Diagrama de bloques sobre el sistema de detección y seguimiento de colibríes y monitoreo de frecuencias en vuelo y pausas en vuelo.

FIG. 2 Ejemplifica las pausas realizadas por el colibrí en diferentes frames. En los intervalos de tiempo $[0; t_a]$ y $[t_b; t_c]$ Se presentan los estados 1 y 3 aleteo superior, entre los intervalos $[t_a; t_b]$ y $t_c; t_d]$ estado 2 y estado 4 aleteo inferior; Finalmente, entre $[td; t_e]$ Se ejemplifica el estado de pausa.

FIG 3. Detalles del funcionamiento de los módulos de detección y clasificación.

FIG 4. Detalle del método utilizado para la medición de frecuencias de vuelo y detección de pausas en vuelo de los colibríes.

DESCRIPCIÓN DETALLADA:

El sistema aquí descrito presenta la metodología empleada para realizar la detección y seguimiento de colibríes, así como otros objetos (especificados por el usuario como entrada de texto), en videos, utilizando estrategias de visión por computador. Adicionalmente, el sistema tiene la capacidad de medir la frecuencia de vuelo y monitorear las pausas realizadas por el colibrí, lo que aumenta la usabilidad del sistema para el análisis de comportamiento animal. El sistema completo se ilustra en la FIG 1. El sistema incluye el hardware y software para el monitoreo y obtención de videos y un módulo para la descomposición en frames de video (sección 1.1), un módulo para la detección de objetos (sección 1.2), un módulo para la clasificación de los objetos detectados (sección 1.3), entre las clases: colibrí y las demás clases definidas por el usuario a través de una entrada de texto, un módulo de seguimiento (sección 1.4) y un módulo de monitoreo de pausas (sección 1.5).

1.1 Interfaz de Video:

El sistema de interfaz de video consiste en una configuración de cámara con punto fijo, La resolución mínima requerida es 800×600 con una velocidad mínima de 500 frames/segundo de grabación y de 30 frames/segundo de reproducción, con el fin de hacer un análisis en cámara lenta. Los frames de video son procesados teniendo un espacio de color RGB y el análisis se realiza sobre los frames de video que son extraídos, procesados por todo el método y eliminados de la memoria para reducir el costo computacional. El algoritmo se corre en una unidad de procesamiento con acceso a GPU 6 Gb, que soporta cuDNN y CUDA 11.

1.2 Módulo de detección de objetos:

El objetivo principal de este módulo consiste en detectar los objetos que se encuentran en la imagen. La detección consiste en determinar la(s) caja(s) (bounding box) que contiene todos los píxeles de los objetos detectados. Una clase particular, puede contener diferentes variaciones a través de las imágenes tales como cambios de pose, de iluminación, deformaciones, entre otros. Sistemas basados en deep learning, como la familia de detectores YOLO [25] son robustos a este tipo de variaciones. Estos sistemas son entrenados en datos etiquetados tanto detección como en clasificación, de esta forma se construye un sistema multi etiqueta para abarcar las dos tareas. Adicionalmente, este algoritmo basado en

redes neuronales artificiales tiene también la característica de ser un modelo pequeño de cálculo rápido lo que lo hace óptimo para el tratamiento de videos. La habilidad de generalización de estos algoritmos es amplia lo que permite utilizar modelos pre-entrenados en una diversidad de tareas. El algoritmo YOLO v8 [25] tiene una entrada de imagen de tamaño 640×640 , una arquitectura *backbone* conformada por capas convolucionales, con capas de pooling intermedias, y capas cuello de botella (bottleneck) de tamaño 1×1 , para reducir el tamaño de parámetros, salidas intermedias de la arquitectura son utilizadas como entrada a la etapa de detección. La salida del detector predice múltiples posibles cajas que envuelven el objeto, para determinar la mejor predicción, a través de un algoritmo denominado *non-maximum supression*, se eliminan aquellas cajas que sobrepasan un porcentaje de solapamiento, dado por el usuario, medido a través de la intersección sobre la unión de dichas cajas.

En este método, nos basamos en este tipo de algoritmos para realizar la detección de objetos en la imagen, partiendo de un entrenamiento global a partir de la base de datos de referencia, caracterizada por tener: 80 clases de objetos, 330.000 imágenes etiquetadas con de la caja que encuadra cada objeto identificado [40] y que entre sus clases se encuentra la detección de la clase colibrí (identificada como clase 14); La optimización del método propuesto se realiza a través de la minimización de la función objetivo [41], detallada en la segunda reivindicación de este documento. Dicha función consta de tres funciones de optimización que al sumarse obtienen la función de pérdida global de la detección:

1. Función objetivo de los bounding boxes:

Detecta la pérdida relacionada a las bounding boxes etiquetadas $(\hat{x}, \hat{y})$ y las predichas (x, y) , así como relaciona el ancho w y alto h de la caja. Esta función tiene una constante de regularización $\lambda_{coord} = 0.5$ utilizada para penalizar aquellas cajas erróneas. B es el número de cajas detectadas y S es el número de celdas en las que se divide la imagen para realizar los cálculos.

2. Función objetivo de la confianza de la predicción de los bounding boxes:

Detecta la pérdida relacionada a la probabilidad de confianza sobre la predicción de cada bounding box. En esta función, C es el puntaje de confianza respecto a la existencia del objeto y $\hat{C}$ es la intersección sobre la unión (coeficiente de Jaccard) entre la caja predicha y la etiquetada. Adicionalmente, esta función penaliza la no existencia de un objeto por medio de la constante de regularización $\lambda_{noobj} = 0.5$.

3. Función objetivo de la predicción de la clase:

Finalmente, la función de pérdida de la clasificación, calcula la suma del error medio cuadrado cuando existe un objeto en la celda, representado por 1_{ij}^{obj} .

Esto es necesario ya que se busca que el modelo no se vea afectado cuando existe una predicción donde ningún objeto es detectado.

La detección de la clase no colibrí se realiza sin tener en cuenta las clases puntuales a las que pertenecen los objetos, ya que se realizará posteriormente una reclasificación; ver FIG 3: Módulo de detección. Esto nos permite utilizar métodos más robustos y específicos en la clasificación y seguimiento de los objetos de interés, que constituyen a los colibríes y otros elementos sugeridos por los usuarios a través de mensajes de texto (text prompts). Ver 1.3: Módulo de Clasificación. Adicionalmente, es posible configurar el uso de diferentes modelos entrenados con diferentes tamaños adecuándose a las necesidades de los futuros usuarios.

1.3 Módulo de Clasificación:

Los algoritmos basados en la familia YOLO son robustos en detectar objetos pero su clasificación tiende a tener un menor rendimiento. En este sistema creamos un módulo de reclasificación lo que nos permite crear un procedimiento adaptable a las necesidades del usuario y a utilizar métodos de clasificación más robustos; ver FIG 3: Módulo de Clasificación. Este módulo cumple dos funciones, la primera (1), consiste en extraer características de las secciones de la imagen dónde se realizó una detección (regiones de interés de la imagen o roi) y de mensajes de texto editables lo que le permitiría ser adaptable a futuras entradas y necesidades de los usuarios. El mensaje de clase base es “hummingbird” (colibrí en inglés), de igual forma, se permite que el usuario ingrese clases adicionales que pueden estar incluidas en sus videos y que pueden variar dependiendo del uso, ej: Flores, alimentadores en plástico, persona, gorrión, etc.

El módulo se basa en un algoritmo transformers de tipo Large Scale Learning Model (LLMs) denominado CLIP [30], si bien los mensajes de texto no son infinitos, el sistema está entrenado con una extensa base de datos que mapea características de texto con características de imagen y a través de la minimización de la distancia, generalmente la distancia del coseno, entre los dos tipos de características bimodales (texto e imagen) es posible determinar la similitud entre ellos. De esta forma es factible realizar una clasificación de los objetos detectados sin la necesidad de hacer un re-entrenamiento y limitando el universo de búsqueda a los objetos (animados o inanimados) de interés.

El procedimiento para realizar la reclasificación propuesta consiste en:

- a) Recolectar a partir de entrada de texto por parte del usuario las clases adicionales de interés.

- b) Extraer características (embeddings) de texto de la palabra colibrí así como de las demás clases añadidas por el usuario.
- c) Recortar la imagen extrayendo las regiones de interés que contienen objetos detectados por el método de la sección 1.2 en el frame actual.
- d) Extraer características de imagen para cada una de las regiones de interés detectadas.
- e) Realizar un análisis a partir del mapeo de la distancia del coseno, presentada en la reivindicación 3, entre las características de texto (embeddings A en la reivindicación) e imagen (embeddings B en la reivindicación) obtenidas. La menor distancia encontrada corresponde a la clase a la que pertenece el objeto.
- f) Repetir el procedimiento en cada uno de los frames de video.

La segunda función (2), consiste en transferir las características extraídas de la imagen para cada región de interés al módulo de seguimiento, ya que con estas será posible realizar la comparación entre los objetos detectados en frames secuenciales.

1.4 Módulo de seguimiento:

El módulo de seguimiento, recibe las características extraídas en la etapa de clasificación, en el primer frame de video estas características son almacenadas, a cada detección se le provee con un nombre perteneciente a la clase y un número, ej: “hummingbird 1”, de forma que cada objeto quede identificado, a continuación se procede a almacenar temporalmente las características del frame anterior a través de un diccionario, teniendo una estructura de tipo $\{key: value\}$, dónde la key representa la clase con su respectivo código de identificación y el value las características extraídas. Para los siguientes frames de videos, se realiza una comparación, a través de la similitud del coseno, presentada en la reivindicación 3, entre las características obtenidas para el frame actual (embeddings A en la reivindicación) y las almacenadas en el diccionario (embeddings B en la reivindicación), al detectar aquellas que son muy cercanas se realiza una actualización de dicho diccionario, reemplazando las características del frame anterior con las del frame actual. En caso de tener una distancia que supera cierto umbral parametrizable, se interpreta que el objeto no existió en el frame anterior y, por lo tanto, se crea una nueva key en el diccionario.

1.5 Medición de Frecuencias de vuelo y Detección de Pausas en Vuelo:

Como ave polinizadora, el estudio del comportamiento del colibrí es de gran importancia en la preservación del balance de diferentes ecosistemas. Es así como en esta invención proponemos un método de caracterización de comportamiento para medir las pausas aéreas que realiza esta ave en su estado de vuelo (ver FIG 2). Para lograr este propósito, a partir de la detección y clasificación realizadas,

proponemos filtrar las detecciones realizadas para cada colibrí identificado a lo largo del video, tomar secuencialmente cada uno de los frames con detecciones y obtener la distancia de Jaccard, presentada en la reivindicación 4, que mide la intersección dividido sobre la unión de dos áreas determinadas (ver FIG 4). En nuestro caso, el área_1 corresponde a la caja que delimita el colibrí identificado en el frame_n y área_2 el corresponde a la caja que delimita la detección del mismo colibrí en el frame_{n-1} . De esta forma es posible obtener un comportamiento que caracteriza *distancia de Jaccard vs tiempo* dónde el tiempo corresponde al tiempo de duración del video. Con esta medición podemos realizar la medición de las frecuencias de vuelo y las pausas en vuelo como se describe a continuación y se profundizan en la reivindicación 10 de este documento:

- A partir de la medición de los cambios de locación de las cajas de detección en los frames consecutivos, y, utilizando, la transformada de fourier de tiempo corto (STFT) es posible medir la distribución de las frecuencias durante una razón de un término de tiempo [22], a partir de esta medición se obtienen los picos de la función de frecuencia lo que corresponde a la cantidad de aleteos del ave en un lapso de tiempo.
- Para medir las pausas en vuelo, los puntos dónde la derivada de la distancia con respecto al tiempo es cercana a cero (0) en un espacio de tiempo prolongado corresponden a los puntos específicos dónde se monitorea la pausa.

MEJORES FORMAS DE LLEVAR A CABO LA INVENCION:

1. Parámetros configurables:

- Source: dirección de memoria dónde se encuentra almacenado el video, en string de texto. Cada directorio se debe separar con slash "/".
- Detection confidence: umbral de probabilidad para tener en cuenta una detección como válida.
- Overlap: Umbral de valor mínimo, con rango entre 0.5 y 0.7 tenido en cuenta en la detección para la etapa de nms.
- Project: Directorio para indicar dónde se almacenará el resultado, en memoria por ejemplo.
- Max cosine distance: umbral máximo para considerar la distancia entre los embeddings al calcular la distancia del coseno. Entre un rango de 0.4 y 1.0.

- Clip labels: Lista de etiquetas de texto configurables sobre las cuáles se realizará la clasificación de los objetos. Almacenadas en un archivo de texto configurable.
- Pause thresholds: Umbral del valor máximo de desviación a tener en cuenta en el cálculo de la derivada para la detección de las pausas. Entre 0.0 y 0.3.

2. Pasos del proceso:

- Paso 1: Extraer embeddings de texto, tal cómo se explica en el módulo de clasificación, Descripción detallada, Sección 1.3.
- Paso 2: Inicializar el modelo de detección, tal como se explica en el módulo de detección, Descripción detallada, Sección 1.2.
- Paso 3: Definir función de métrica de distancia coseno. Utilizada tanto en el módulo de clasificación, como en el módulo de seguimiento, Descripción detallada Secciones 1.3 y 1.4
- Paso 4: Extraer los frames del video, tal como se explica en la Descripción Detallada, Sección 1.1.
- Paso 5: Análisis por cada uno de los frames de video disponibles.
for frame in frames:
 - Paso 6: Obtener detecciones, procesando el frame por el módulo de detección.
 - Paso 7: Obtener regiones de interés a partir de las detecciones, para ser procesadas por el módulo de clasificación.
 - Paso 8: Obtener embeddings de imagen de las regiones de interés como se detalla en el módulo de clasificación.
 - Paso 9: Contrastar los embedding de texto con los de imágenes a través de la distancia del coseno. Para obtener la clasificación del objeto.
 - Paso 10: Actualización del módulo de seguimiento. Ver Descripción Detallada, Sección 1.4.
- Paso 11: Se obtiene el coeficiente de Jaccard en los frames consecutivos, tal como se explica en la Sección 1.5 de la Descripción Detallada.
- Paso 12: Se obtiene la transformada de tiempo corto de fourier, tal como se detalla en la Sección 1.5 de la Descripción Detallada.
- Paso 13: Se obtiene la primera componente de frecuencia y se aplica el filtro de savgol para suavizar la señal.
- Paso 14: Se cuentan los picos de frecuencia de la señal que corresponden a la frecuencia de aleteo.
- Paso 15: Se obtiene la derivada de la función del coeficiente de Jaccard.
- Paso 16: Se marcan las pausas detectadas.

3. secuencia de etapas del método

A continuación, se presenta a manera de pseudocódigo el módulo principal de la invención, presentando el paso a paso de la invención antes descrita. Adicionalmente, se presenta un diagrama de bloques con el proceso. las etapas a seguir por el método implementado por computador son:

__init__:

text_labels : etiquetas de texto personalizables
model_classifier: Definición clip classifier, recibe text_labels
metric: Model: Definición métrica de distancia (cosine_distance)
yolo_engine: Se define el modelo de detección.
source: se define la dirección del proyecto
tracker: inicializar el diccionario
pause_th: umbral para regularizar el monitoreo de pausa

process:

frames: se extraen frames de video.
for frame **in** frames:
 se procesa la imagen por la detección
 boxes, confianza, classes =yolo_engine(frame)
 se obtienen regiones de interés, se extraen características y
 se hace la clasificación
 for box **in** boxes:
 roi = frame[box]
 clase, features = model_classifier(roi)
 actualizar el sistema de seguimiento
 tracker.update([box, confianza, clase, features])
Se calcula la distancia de jaccard entre frames consecutivos
for index **in** range(len(tracker)):
 box_frame_n = tracker[box][index]
 box_frame_n1 = tracker[box][index + 1]
 iou.append(jaccard_distance(box_frame_n , box_frame_n1)

Se calcula la STFT de la distancia de jaccard
Sxx = signal.spectrogram (iou)
Se obtiene la primera componente de frecuencia
high_freq = Sxx[0,:]
Se aplica un filtro para suavizar la señal
filter = signal.savgol_filter(high_freq)
Se detectan los picos de la señal filtrada
peaks = signal.find_peaks(filter)

Se calcula la derivada de la distancia de jaccard

```
diff_iou = derivate(iou)
```

Se obtiene el valor medio del valor absoluto de la derivada

```
mean = abs(diff_iou).mean()
```

Se digitaliza la derivada obteniendo las regiones planas de la función teniendo un umbral máximo (pause_th) de cambio.

```
for value in diff_iou:
```

```
    value = abs(value)
```

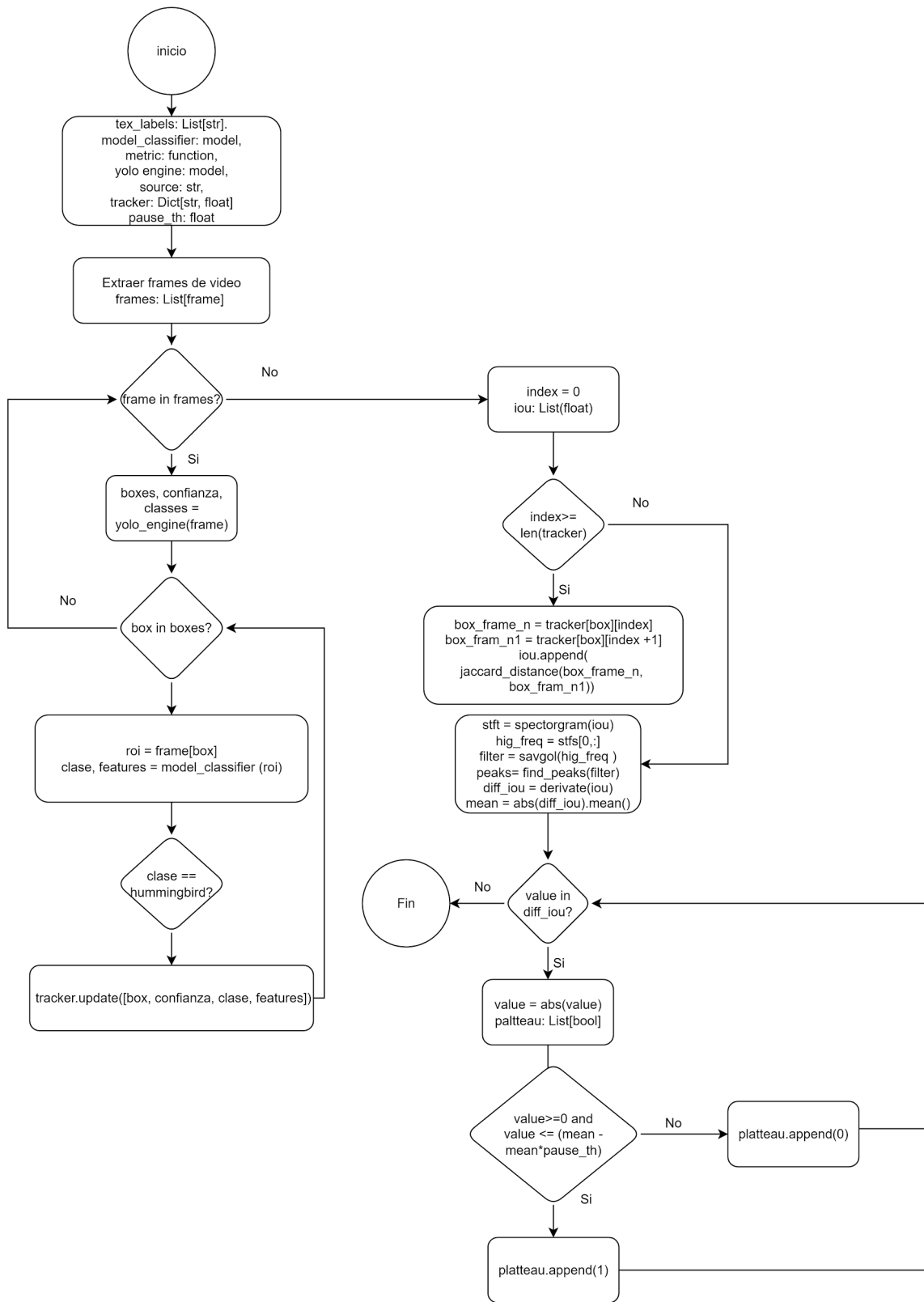
```
    if value >= 0 and value <= (mean - mean*pause_th):
```

```
        platteau.append(1)
```

```
    else:
```

```
        platteau.append(0)
```

Se marcan las pausas cuando platteau = 1



REIVINDICACIONES:

1. Un método implementado por computador en el cual el procesador ejecuta etapas para monitorear y detectar pausas en vuelo de colibríes, caracterizado por las etapas de:
 - A. Una primera etapa de captura de videos en condiciones naturales a traves de un sensor CCD:
 - Grabar videos de Colibries a una velocidad de 500 fps en una estación de video.
 - Extraer los fotogramas del video para identificar las pausas en los vuelos de colibríes. Enviar una primera notificación en un cuadro de texto solicitando ingresar los objetos de interés a detectar por el usuario, que incluyen “hummingbird” y “flower”.
 - Envíar una segunda notificación en la cual se determina la eliminación de los fotogramas ya procesados para optimizar recursos computacionales.
 - Envíar a la nube los fotogramas procesados para alimentar la red de entrenamiento.
 - B. Una segunda etapa de detección de objetos en el video a partir de un algoritmo de detección ejecutado por un procesador que incluye redes convolucionales con pesos pre-entrenados en imágenes con 80 clases diferentes, donde los las imágenes de colibríes:
 - Inicializar el algoritmo se carga en memoria el modelo pre entrenado de detección.
 - Definir parámetros de optimización como umbrales de confianza y de intersección para determinar la selección de las cajas de delimitación finales de los objetos en 0.7.
 - Procesar cada uno de los frames de video para localizar objetos de interés en la imagen, que serán clasificados en una siguiente etapa.
 - Las cajas de delimitación (bounding boxes) encontradas se utilizan como entrada tanto en la etapa de clasificación como en la de seguimiento de objetos.
 - C. Una tercera etapa de clasificación de los colibríes de acuerdo a la similitud de características utilizando la distancia del coseno a partir del conjunto entre pares texto - imagen y corrigiendo las detecciones obtenidas en la etapa anterior (redes convolucionales), en la que el procesador se configura para que:
 - Al inicializar el algoritmo se carga el modelo pre entrenado de clasificación.

- Se extraen las características del texto ingresado por el usuario como objetos de interés.
- Una vez realizada la detección de objetos, se extrae la región de interés para obtener la sección de la imagen donde se obtendrán las características del objeto.
- Se comparan las características de texto con las características de la imagen, utilizando la distancia del coseno.
- En esta etapa se realiza también una sub-etapa, en la cual se realiza una fase de rechazo a la identificación de objetos diferentes a los colibríes y cuya definición inicial se realiza a través de entradas de texto personalizables por el usuario utilizando ingeniería de texto (prompt engineering);

D. Una cuarta etapa de seguimiento de cada colibrí durante el vuelo asignando un código de identificación individual a partir de la comparación entre características (embeddings) de frames secuenciales, en la que el procesador esta configurado para:

- obtener el seguimiento del objeto(s) de interés, a partir de las cajas de detección y las características de la imagen definidas en la primera notificación.
- Crear identificadores para los objetos encontrados en el primer fotograma de video.
- A partir del siguiente fotograma, el fotograma actual se convierte en la identificación anterior para comparar características de imagen entre el cuadro anterior, el actual y el siguiente, con la técnica de la frecuencia del coseno.
- Si la distancia del coseno supera un umbral dado, por defecto, 0.4, entonces, se asume que el objeto identificado es el mismo que se había almacenado en un principio. De lo contrario, se procesa como un nuevo objeto.

E. Una quinta etapa de monitoreo de la frecuencia de aleteo y las pausas aéreas realizadas por cada colibrí, el procesador se configura para:

- Seleccionar los colibríes encontrados y calcular la distancia de Jaccard entre las cajas limitantes (bounding boxes) que tienen el mismo id de identificación dado por el módulo de seguimiento de fotogramas subsiguientes. Cada caja se compara con la del fotograma anterior, con el fin de medir el movimiento realizado por el colibrí a lo largo del vídeo.
- A partir de la señal generada por el coeficiente se calculan:
 - i. La frecuencia de aleteo, obtenida a partir de:
 - La transformada de Fourier de tiempo corto (STFT).

- El suavizado de la frecuencia principal aplicando un filtro savgol.
 - El conteo de picos de la frecuencia. Que corresponden a la frecuencia de aleteo.
- ii. Las pausas en vuelo del colibrí, obtenida a partir de:
- Cálculo de la derivada de la señal con respecto al tiempo.
 - Detección de regiones constantes a partir de la derivada, tomando valores localizados entre cero y la media de la señal

2. Un método de monitoreo de pausas en vuelo de colibríes, de acuerdo con la reivindicación 1, en el cual la función objetivo del problema de optimización para la detección de objetos, se divide en tres partes:

$$L_{loss} = L_{bbox} + L_{bbox_{score}} + L_{class_{score}}$$

Dónde:

- L_{bbox} : Función objetivo para encontrar las coordenadas de la caja (bounding box) que rodea el objeto de interés.

$$L_{bbox} = \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{obj} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2] + \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{obj} [(\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2]$$

- $L_{bbox_{score}}$ Función objetivo para determinar la confianza de la detección.

$$L_{bbox_{score}} = \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{obj} (C_i - \hat{C}_i)^2 + \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{noobj} (C_i - \hat{C}_i)^2$$

- $L_{class_{score}}$: Función objetivo para determinar la confianza de la clasificación del objeto.

$$L_{class_{score}} = \sum_{i=0}^{S^2} 1_{ij}^{obj} \sum_{C \in classes} (\rho_i(C) - \hat{\rho}_i(C))^2$$

Dónde:

- 1_{ij}^{obj} : es 1 si la caja j se encuentra en la celda i ; 0 en otro caso.
- (x, y) y (w, h) : son las coordenadas, la altura y el ancho de la caja predicha.

- $(\hat{x}, \hat{y})$ y $(\hat{w}, \hat{h})$: son las coordenadas, la altura y el ancho de la caja etiquetada como verdadera.
- λ_{coord} y λ_{noobj} : son constantes de regularización.
- B : es el número de cajas encontradas.
- S : Es el número de celdas en la grilla sobre la que se divide la imagen.
- C : Es el puntaje de confianza modulado por la presencia del objeto.
- $\hat{C}$: Es el coeficiente IOU de la caja predicha con la caja etiquetada.
- $\rho_i(C)$: corresponde a la clase predicha.
- $\hat{\rho}_i(C)$: corresponde a la clase etiquetada como verdades.

3. Un método de monitoreo de pausas en vuelo de colibríes, de acuerdo con la reivindicación 1, en donde la distancia del coseno de la clasificación se calcula con la siguiente fórmula:

$$similarity(A, B) = \cos(\theta) = \frac{A \cdot B}{||A|| ||B||}$$

Dónde:

- θ es el ángulo entre los vectores
- $A \cdot B$ es el producto punto entre los vectores A y B

$$A \cdot B = \sum_{i=1}^n A_i B_i = A_1 B_1 + A_2 B_2 + \dots + A_n B_n$$

- $||A||$ representa la norma L2:

$$||A|| = \sqrt{\sum_{i=0}^n |A_i|^2} = \sqrt{A_1^2 + A_2^2 + \dots + A_3^2}$$

En esta patente utilizamos la distancia del coseno en dos situaciones:

- Clasificación: Cuando los vectores A y B representan los embeddings de texto y los embeddings de imagen que se busca comparar para la clasificación.
- Seguimiento: Cuando los vectores A y B representan los embeddings de imagen almacenados en el diccionario de seguimiento de frames anteriores y los embeddings de imagen obtenidos en el frame actual.

4. Un método de monitoreo de pausas en vuelo de colibríes, de acuerdo con la reivindicación 1, en donde la distancia de Jaccard, de la etapa de monitoreo así como su derivada, miden la similitud entre dos conjuntos de muestras, y su desplazamiento en el tiempo, y se calculan con la siguiente ecuación:

$$sim_J(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

y determinan el movimiento bounding box de la detección de un mismo objeto a través del tiempo, arrojando un conjunto de datos en los cuales se

determina si la unión es cercana a uno (1.0) para los objetos que no tuvieron movimiento y están detenidos en una misma posición.

Finalmente, para medir esta detención se realiza el cálculo de la derivada entre la distancia de Jaccard y el tiempo $\frac{d(sim_j(A, b))}{d(t)}$. Calculando en estado constante la tendencia a cero de la derivada.

5. Un método de monitoreo de pausas en vuelo de colibríes, de acuerdo con la reivindicación 1, en donde la base de datos consiste en 15 vídeos tomados en condiciones naturales a diferentes horas del día y tipos de iluminación variables de acuerdo con las condiciones climáticas, grabados a una velocidad de 500 fps y una velocidad de reproducción de 30 fps. Cada video genera una etiqueta corroborada por el usuario en donde se detectan las pausas generadas por el (los) colibrí(es) presente(s) y clasificar aquellos en los que hay más de un colibrí en el video.
6. Un método de monitoreo de pausas en vuelo de colibríes, de acuerdo con la reivindicación 1, que además comprende una estación base caracterizada por una unidad gráfica de procesamiento (GPU) Nvidia T4 con 16 Gb de RAM, una CPU Intel(R) Xeon(R) con 2 núcleos, 12 Gb de RAM y 2 Tb de espacio en disco y cuyo proceso corresponde a:

Recibir la señal enviada por una cámara de punto fijo, con una velocidad de grabación de 30 fps.

Obtener y almacenar el video en la carpeta de destino seleccionado.

Monitorear las pausas en vuelo de forma offline a través del método aquí propuesto.

Almacenar un archivo de texto, comma-separated values file (.csv), que contenga 7 columnas, definidas como: i) frame: el número de frame dónde se genera una detección, ii) bbox: el bounding box con la detección del objeto, iii) class: la clase asignada por el método de clasificación, iv) track: el número de identificación dado por el módulo de seguimiento, v) iou: la distancia de Jaccard entre el frame actual y el siguiente, vi) dy/dx: la derivada de la distancia de Jaccard con respecto al tiempo y, en caso de monitorear una pausa en ese frame, vii) pause: una variable booleana definida como "True" identificando una posible pausa.

7. Un método de monitoreo de pausas en vuelo de colibríes, de acuerdo con la reivindicación 1, que comprende un módulo de procesamiento de video el

cual ejecuta un conjunto de instrucciones implementadas por computador para la captura de un intervalo de video; descomposición en frames de video, validación del tamaño (alto y ancho) de frame así como de la velocidad de grabación medida en frames por segundo mínimas necesarias para el análisis.

8. Un método de monitoreo de pausas en vuelo de colibríes, de acuerdo con la reivindicación 1, que comprende un método para la detección de pausas en vuelos de colibríes caracterizado por:

Un módulo de detección de objetos a partir de un modelo de redes neuronales convolucionales, con extracción de características de capas intermedias, que incluye:

Un primer entrenamiento realizado en la base de datos benchmark, con 330.000 imágenes, 1.5 millones de objetos identificados y 80 categorías, que contiene entre sus clases (identificada como 14) la detección de colibríes y

Una optimización de la detección mediante un re-entrenamiento realizando la transferencia de aprendizaje y sintonización de pesos priorizando la detección de colibríes vs no colibríes.

9. Un método de monitoreo de pausas en vuelo de colibríes, de acuerdo con la reivindicación 1, que comprende etapas para la clasificación de objetos basados en la obtención de características a partir de un modelo transformers multimodal (texto - imagen). Este módulo se caracteriza por:

Tener una clase base de clasificación: colibrí;

Recibir clases adicionales como entrada de texto por parte del usuario, por ejemplo: flores, hojas, alimentadores de aves.

Recortar las imágenes en regiones de interés obtenidas por las cajas de detección;

Obtener características de texto de las clases de interés;

Obtener características de imagen de las regiones de interés y

Comparar a través de la distancia del coseno las características texto - imagen, minimizando la distancia para determinar la clase a la que pertenece cada objeto.

10. Un método de cálculo de las frecuencias de vuelo de colibríes, de acuerdo con la reivindicación 1, que contempla la etapa de seguimiento del vuelo de los colibríes. Este análisis de frecuencia-tiempo se calcula como una colección de múltiples transformadas discretas de fourier a través de ventanas de intervalos de la señal, representadas por la función dada a continuación [23].

$$X_{STFT}[m, n] = \sum_{k=0}^{L-1} x[k]g[k - m]e^{-j2\pi nk/L}$$

Dónde:

- $x[k]$: Corresponds to the signal
- $g[k]$: denota la ventana de análisis.
- L : denota la longitud de $g[k]$

Y un método de monitoreo de pausas en vuelo de colibríes, de acuerdo con la reivindicación 1, que comprende etapas para realizar el seguimiento del vuelo de los colibríes durante un intervalo de tiempo a través de los frames de video utilizando extracción de embeddings. Estas características (embeddings) se retroalimentan de los obtenidos en la etapa de clasificación, algoritmo transformers bimodal (texto-image) tomando únicamente las características de imagen, las cuales son almacenadas en un diccionario teniendo como llave (key) el track de identificación y la clase del elemento. Seguidamente se comparan entre frames subsiguientes a través de la distancia del coseno de los embeddings almacenados y los embeddings del frame actual.

FIGURAS

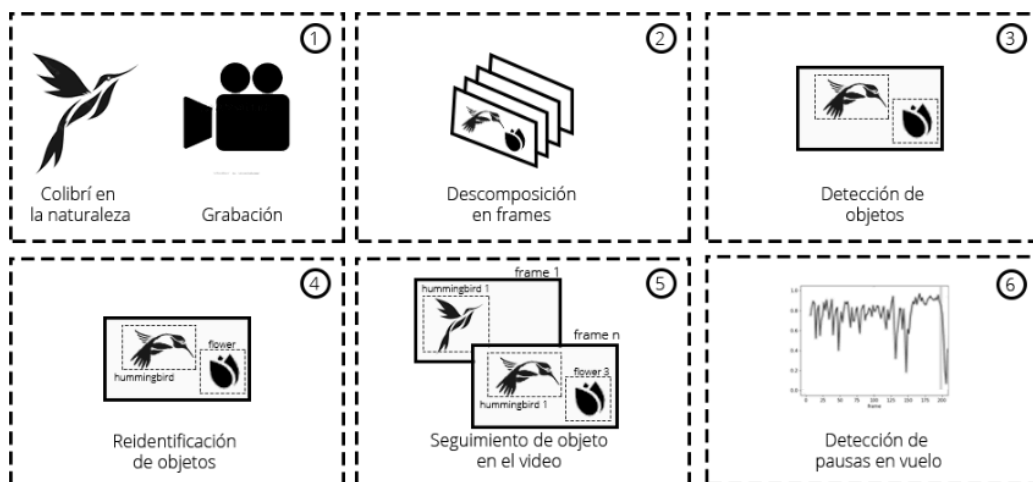


FIG 1. Diagrama de bloques sobre el sistema de detección y seguimiento de colibríes y monitoreo de pausas en vuelo.

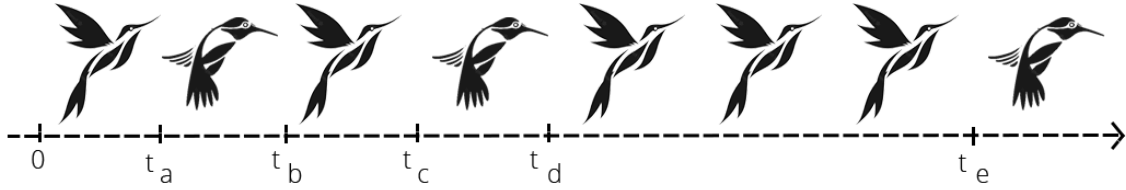
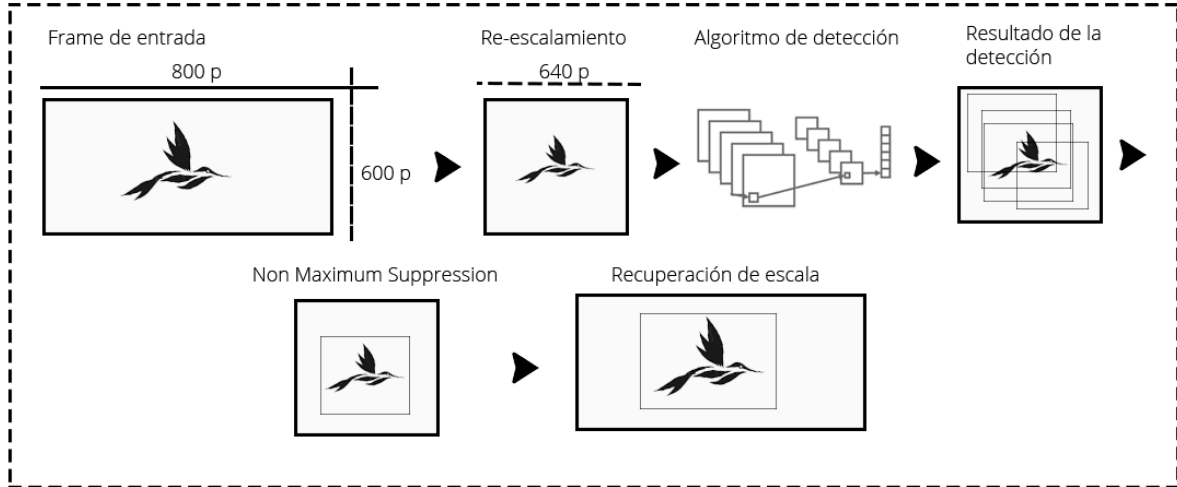


FIG 2. Ejemplifica el vuelo realizado por el colibrí en diferentes estados de tiempo. En los intervalos de tiempo $[0; t_a]$ y $[t_b; t_c]$ Se presenta estado de aleteo superior, entre los intervalos $[t_a; t_b]$ y $[t_c; t_d]$ estado de aleteo inferior; Entre $[t_d; t_e]$ estado de pausa.

Módulo de detección



Módulo de Clasificación

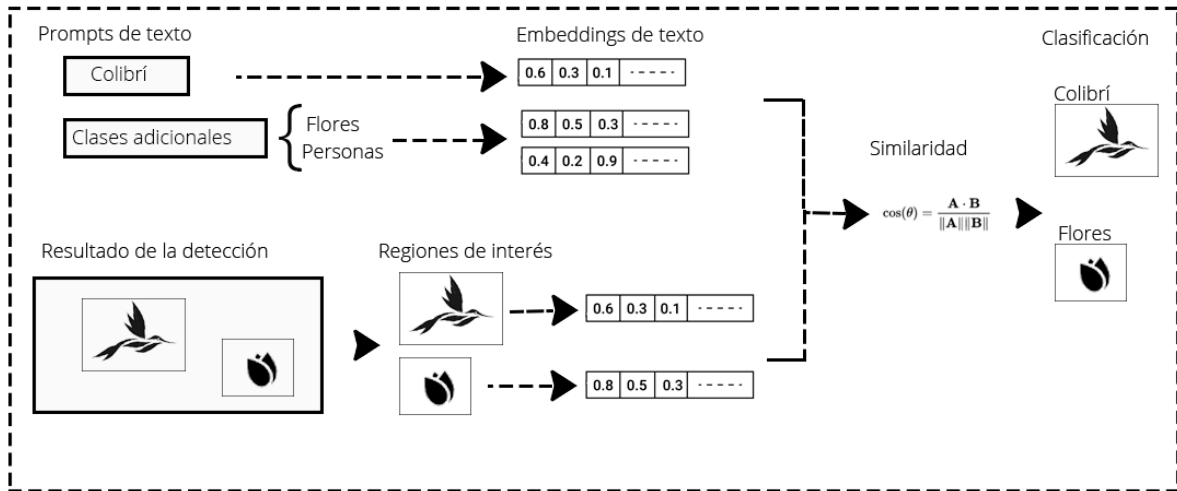


FIG 3. Detalles del funcionamiento de los módulos de detección y clasificación.

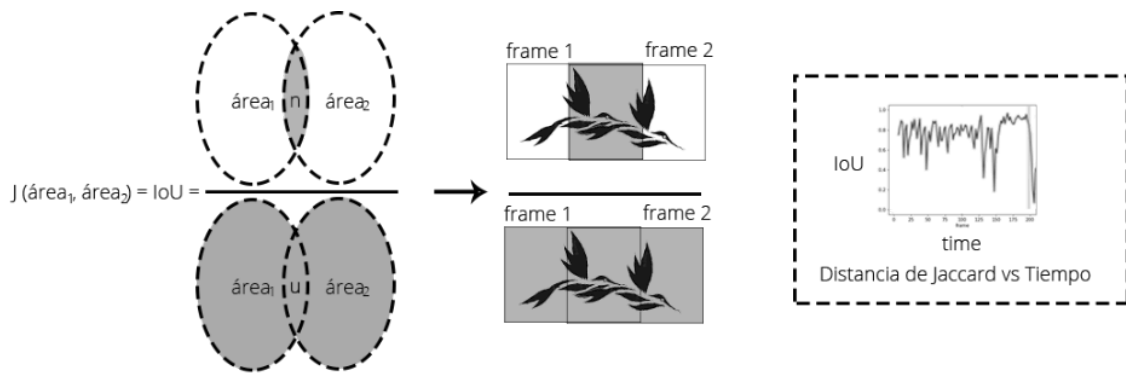


FIG 4. Detalle del método utilizado para la detección de pausas en vuelo de los colibríes.

REFERENCIAS:

- [1] Rico-Guevara A., Hurme K.J., Elting R., Russel A.L.I; Bene “fit” assessment in pollination coevolution: mechanistic perspectives on hummingbird bill–flower matching; Integrative and Comparative Biology; 2021; Volume 61; issue 2; pp 681-695.
- [2] Gutierrez Zamora A., Carrillo Chica E., Rojas Nossa, S.; Guía Ilustrada de los Colibríes de la Reserva Natural Río Ñambí; Ecotono; 2004.
- [3] Altshuler, D. L. and Dudley, R. (2002). The ecological and evolutionary interface of hummingbird flight physiology. *Journal of Experimental Biology*, 205 (16):2325–2336.
- [4] Skandalis, D. A., Segre, P. S., Bahlman, J. W., Groom, D. J. E., Welch, K. C., Witt, C. C., McGuire, J. A., Dudley, R., Lentink, D., and Altshuler, D. L. (2017). The biomechanical origin of extreme wing allometry in hummingbirds. *Nature Communications*, 8(1):1047.
- [5] Warrick, D. R., Tobalske, B. W., and Powers, D. R. (2009). Lift production in the hovering hummingbird. *Proceedings of the Royal Society B: Biological Sciences*, 276(1674):3747–3752.
- [6] Altshuler, D., Stiles, F., and Dudley, R. (2004). Of Hummingbirds and Helicopters: Hovering Costs, Competitive Ability, and Foraging Strategies. *The American Naturalist*, 163(1):16–25.

- [7] Tobalske, B. W., Warrick, D. R., Clark, C. J., Powers, D. R., Hedrick, T. L., Hyder, G. A., and Biewener, A. A. (2007). Three-dimensional kinematics of hummingbird flight. *Journal of Experimental Biology*, 210(13):2368–2382.
- [8] Nan, Y., Peng, B., Chen, Y., and McGlinchey, D. (2019). From Studying Real Hummingbirds to Designing Hummingbird-Like Robots—A Literature Review. *IEEE Access*, 7:131785–131804.
- [9] Tu, Z., Fei, F., Zhang, J., and Deng, X. (2020). An At-Scale Tailless Flapping-Wing Hummingbird Robot. I. Design, Optimization, and Experimental Validation. *IEEE Transactions on Robotics*, 36(5):1511–1525.
- [10] Wilcox, S. C. and Clark, C. J. (2022). Sexual selection for flight performance in hummingbirds. *Behavioral Ecology*, 33(6):1093–1106.
- [11] Zhao, W., Zhang, Z., Mirjalili, S., Wang, L., Khodadadi, N., and Mirjalili, S. M. (2022). An effective multiobjective artificial hummingbird algorithm with dynamic elimination-based crowding distance for solving engineering design problems. *Computer Methods in Applied Mechanics and Engineering*, 398:115223.
- [12] Jensen, J. H., Dyre, J. C., and Hecksher, T. (2024). Universal wing- and fin-beat frequency scaling. *PLOS ONE*, 19(6):e0303834.
- [13] Martínez, F., Manzanera, A., and Romero, E. (2015a). Automatic analysis and characterization of the hummingbird wings motion using dense optical flow features. *Bioinspiration & Biomimetics*, 10(1):016006.
- [14] Dakin, R., Segre, P. S., Straw, A. D., and Altshuler, D. L. (2018). Morphology, muscle capacity, skill, and maneuvering ability in hummingbirds. *Science*, 359(6376):653–657.
- [15] Stiles, F. G. (1971). Time, Energy, and Territoriality of the Anna Hummingbird (*Calypte anna*). *Science*, 173(3999):818–821.
- [16] Inouye, D. W., Calder, W. A., and Waser, N. M. (1991). The Effect of Floral Abundance on Feeder Censuses of Hummingbird Populations. *The Condor*, 93(2):279–285.
- [17] Chai, P., Altshuler, D. L., Stephens, D. B., and Dillon, M. E. (1999). Maximal Horizontal Flight Performance of Hummingbirds: Effects of Body Mass and Molt. *Physiological and Biochemical Zoology*, 72(2):145–155.
- [18] Clark, C. J. and Mistick, E. A. (2018). Kinematic control of male Allen's Hummingbird wing trill over a range of flight speeds. *Journal of Experimental Biology*, page jeb.173625.

- [19] Chai, P. (1997). Hummingbird Hovering Energetics During Moulting of Primary Flight Feathers. *Journal of Experimental Biology*, 200(10):1527–1536.
- [20] Díaz-Salazar, A. F., Garzón-Agudelo, F., Smiley, A., Cadena, C. D., and Rico-Guevara, A. (2023). Winging it: Unveiling how hummingbirds alter their flying kinematics during molt. *bioRxiv*.
- [21] Greenewalt, C. H. (1960). The Wings of Insects and Birds as Mechanical Oscillators. 104(6):605–611.
- [22] Steen, R. (2014). The use of a Low Cost High Speed Camera to Monitor Wingbeat Frequency in Hummingbirds (Trochilidae). *Ardeola*, 61(1):111.
- [23] Sotavalta, O. (1952). Flight-Tone and Wing-Stroke Frequency of Insects and the Dynamics of Insect Flight. *Nature*, 170(4338):1057–1058.
- [24] Xian, Y., Lampert, C. H., Schiele, B., and Akata, Z. (2019). Zero-Shot Learning—A Comprehensive Evaluation of the Good, the Bad and the Ugly. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(9):2251–2265.
- [25] Ultralytics YOLOv8 Doc, <https://docs.ultralytics.com/>
- [26] Patil, S. (2024). Open clip. Accessed: 2024-08-15.
- [27] Nguyen, T.-D., Vu, A.-K. N., Nguyen, N.-D., Nguyen, V.-T., Ngo, T. D., Do, T.-T., Tran, M.-T., and Nguyen, T. V. (2024). The Art of Camouflage: Few-Shot Learning for Animal Detection and Segmentation. *IEEE Access*, 12:103488–103503.
- [28] Jiang, P., Ergu, D., Liu, F., Cai, Y., and Ma, B. (2022). A Review of Yolo Algorithm Developments. *Procedia Computer Science*, 199:1066–1073.
- [29] Ma, D. and Yang, J. (2022). YOLO-Animal: An efficient wildlife detection network based on improved YOLOv5. In 2022 International Conference on Image Processing, Computer Vision and Machine Learning (ICICML), pages 464–468, Xi'an, China. IEEE.
- [30] Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I. (2021). Learning Transferable Visual Models From Natural Language Supervision. Version Number: 1.
- [31] Shin, G., Albanie, S., and Xie, W. (2023). Zero-shot Unsupervised Transfer Instance Segmentation. Version Number: 1.

[32] Wu, C. and Maji, S. (2022). How well does CLIP understand texture? Version Number: 2.

[33] Breslav, M., Fuller, M. W., and Betke, M. (2012). Vision system for wingbeat analysis of bats in the wild. In Proceedings of the Workshop on Visual Observation and Analysis of Animal and Insect Behavior (VAIB 2012), Tsukuba, Japan. Citeseer.

[34] Ling, H., McIvor, G. E., Nagy, G., MohaimenianPour, S., Vaughan, R. T., Thornton, A., and Ouellette, N. T. (2018). Simultaneous measurements of three-dimensional trajectories and wingbeat frequencies of birds in the field. *Journal of The Royal Society Interface*, 15(147):20180653.

[35] Serrano, S. A., Benítez-Jimenez, R., Núñez-Rosas, L., Del Coro Arizmendi, M., Greeney, H., Reyes-Meza, V., Morales, E., and Escalante, H. J. (2018). Automated Detection of Hummingbirds in Images: A Deep Learning Approach. In Martínez-Trinidad, J. F., Carrasco-Ochoa, J. A., Olvera-López, J. A., and Sarkar, S., editors, *Pattern Recognition*, volume 10880, pages 155–166. Springer International Publishing, Cham. Series Title: Lecture Notes in Computer Science.

[36] Weinstein, B. G. (2015). Integrating motion video detection and ecological monitoring. *Methods in Ecology and Evolution*, 6(3):357–362.

[37] Weinstein, B. G. (2018). Scene-specific convolutional neural networks for video-based biodiversity detection. *Methods in Ecology and Evolution*, 9(6):1435–1441.

[38] Rico-Guevara, A. and Mickley, J. (2017). Bring your own camera to the trap: An inexpensive, versatile, and portable triggering system tested on wild hummingbirds. *Ecology and Evolution*, 7(13):4592–4598.

[40] Lin, Tsung-Yi Maire, Michael, Belongie, Serge Hays, James, Perona, Pietro Ramanan, Deva Dollár, Piotr and Zitnick C. Lawrence; Microsoft COCO: Common Objects in Context; *Computer Vision -- ECCV 2014*, pp 740 - 755, 2014.

[41] Iffat Zafar; Giounona Tzanidou, Richard Burton, Nimesh Patel and Leonardo Araujo; *Hands-On Convolutional Neural Networks with TensorFlow*; Packt Publishing Ltd, ISBN 978-1-78913-033-1, Object Detection and Segmentation, 2018.