

Universitat Oberta
de Catalunya

Carlos Romero Matarin

Máster universitario en
Ciencia de Datos
Ecology, Ecological Networks
and Deep Learning

Tutor/a de TF

Bernat Bas Pujols

22/12/2024



Esta obra está sujeta a una licencia de Reconocimiento-
NoComercial-SinObraDerivada [3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

Ficha del Trabajo Final

Título del trabajo:	Detección y Análisis de Huellas de Tortugas en Imágenes utilizando Deep Learning
Nombre del autor/a:	Carlos Romero Matarin
Nombre del Tutor/a de TF:	Bernat Bas Pujols
Nombre del/de la PRA:	
Fecha de entrega:	
Titulación o programa:	Máster universitario en Ciencia de Datos
Área del Trabajo Final:	Ecology, Ecological Networks and Deep Learning
Idioma del trabajo:	Castellano
Palabras clave	<p>Algunas de las palabras clave del proyecto serían:</p> <ul style="list-style-type: none"> • Aprendizaje profundo • YOLO (You Only Look Once) • Detección de huellas • Conservación de tortugas • Análisis de imágenes • U-NET
Resumen del Trabajo	
<p>Este estudio desarrolla un sistema automatizado fundamentado en modelos de aprendizaje profundo avanzados, tales como YOLOv8 y U-Net, con el objetivo de identificar y segmentar las huellas de tortugas en imágenes de su entorno natural. El propósito es perfeccionar las estrategias de conservación a través de instrumentos tecnológicos que potencien la exactitud y eficacia del monitoreo.</p> <p>YOLOv8 se distingue por su habilidad para realizar detección en tiempo real rápida y precisa, lo que resulta idóneo para contextos dinámicos. La funcionalidad de U-Net se potencia mediante segmentaciones meticulosas, lo que simplifica el análisis granular de las áreas ocupadas. La incorporación de ambos modelos posibilita una estrategia híbrida que logra un equilibrio entre velocidad y precisión, optimizando los procedimientos de identificación</p>	

y análisis.

El sistema sugerido reemplaza los métodos manuales convencionales, lo que resulta en una reducción de costos y un incremento en la uniformidad de los resultados. Adicionalmente, se enfrentan retos técnicos tales como iluminación variable y fondos complejos a través de técnicas avanzadas de preprocesamiento, tales como la mejora del contraste y el incremento de datos.

Los hallazgos demuestran que YOLOv8 y U-Net demuestran eficacia en contextos reales, facilitando la toma de decisiones basadas en datos consistentes. Esta solución no solo optimiza las iniciativas de conservación, sino que además establece un precedente para la aplicación de inteligencia artificial en la supervisión medioambiental. El sistema favorece la salvaguarda de especies amenazadas y fomenta la utilización sostenible de ecosistemas marinos, evidenciando la capacidad del aprendizaje profundo para abordar cuestiones complejas en biodiversidad y ecología.

Abstract

This study develops an automated system based on advanced deep learning models, such as YOLOv8 and U-Net, aimed at identifying and segmenting turtle tracks in images of their natural environment. The purpose is to enhance conservation strategies through technological tools that improve the accuracy and efficiency of monitoring.

YOLOv8 stands out for its ability to perform rapid and precise real-time detection, making it ideal for dynamic contexts. U-Net's functionality is enhanced by meticulous segmentations, simplifying the granular analysis of occupied areas. The integration of both models enables a hybrid strategy that balances speed and precision, optimizing identification and analysis processes.

The proposed system replaces conventional manual methods, resulting in cost reductions and increased uniformity in results. Additionally, technical challenges such as variable lighting and complex backgrounds are addressed through advanced preprocessing techniques, including contrast

enhancement and data augmentation.

Findings show that YOLOv8 and U-Net are effective in real-world contexts, facilitating decision-making based on consistent data. This solution not only optimizes conservation initiatives but also sets a precedent for the application of artificial intelligence in environmental monitoring. The system supports the protection of endangered species and promotes the sustainable use of marine ecosystems, demonstrating the potential of deep learning to tackle complex issues in biodiversity and ecology.

Resumen (abstract)

El propósito de la presente investigación es desarrollar un sistema automatizado que utiliza modelos avanzados de aprendizaje profundo, particularmente YOLOv8 y U-Net, para la detección y segmentación exacta de las huellas de tortugas en fotografías capturadas en su ambiente natural. El objetivo principal de este estudio es optimizar las estrategias de conservación y análisis de las tortugas marinas, ofreciendo instrumentos tecnológicos innovadores que faciliten un seguimiento más eficaz y minucioso.

El modelo YOLOv8 ha sido seleccionado debido a su habilidad para efectuar detecciones rápidas y exactas en tiempo real, un atributo indispensable en entornos naturales dinámicos y en constante cambio. Además, U-Net enriquece esta funcionalidad mediante una segmentación minuciosa de las huellas, lo cual facilita la identificación de áreas específicas con gran precisión y la obtención de información tanto cualitativa como cuantitativa de las mismas. La integración de ambos métodos constituye una solución integral que optimiza la eficacia en el análisis de las huellas, abordando tanto la identificación inmediata como el análisis exhaustivo.

Este sistema automatizado sustituye los procedimientos convencionales que se llevan a cabo de manera manual, logrando mejoras significativas en la eficiencia operativa y en la exactitud de los datos obtenidos. Además de agilizar la recopilación de información, el sistema asegura una mayor uniformidad en los resultados, lo que resulta en decisiones más fundamentadas y fundamentadas en datos de alta calidad.

La iniciativa se centra igualmente en superar los desafíos vinculados a las condiciones de iluminación fluctuantes y los fondos complejos presentes en los entornos naturales. Para lograr esto, se lleva a cabo una investigación sobre la implementación de técnicas avanzadas de preprocesamiento, tales como la mejora del contraste en imágenes nocturnas. Asimismo, se examinan métodos de ampliación del dataset mediante la utilización de datos sintéticos generados mediante técnicas de incremento de datos y modelos generativos adversarios.

Adicionalmente, el estudio contempla la incorporación de ambos modelos en un sistema híbrido, en el que la celeridad de detección de YOLOv8 se combina con la precisión de segmentación de U-Net, optimizando así el rendimiento global del sistema. Este método no solo consolida su aplicación en proyectos de conservación, sino que también establece un precedente para la implementación de tecnologías de visión por computadora en el campo de la biología de la conservación y otros contextos ambientales.

En última instancia, esta propuesta favorece la conservación de las tortugas marinas, facilita la monitorización de sus patrones de movimiento y amplía el conocimiento sobre el uso de la inteligencia artificial para tratar problemas complejos en el ámbito de la ecología y la biodiversidad.

Índice

1. Introducción	6
1.1. Contexto y Justificación del Trabajo	6
1.2. Motivación Personal	8
1.3. Objetivos del Trabajo.....	8
1.4. Descripción de la metodología utilizada en el desarrollo del proyecto	9
1.5. Planificación del Trabajo.....	11
1.6. Impacto en Sostenibilidad, Ético-social y de Diversidad	12
1.6.1 Impacto en Sostenibilidad	12
1.6.2 Impacto Ético-Social	13
1.6.3 Impacto en Diversidad.....	13
2. Marco Teórico	14
2.1. Técnicas de Visión por Computadora para la Conservación.....	14
2.2. Aprendizaje Profundo en la Conservación	15
2.3. Importancia de la Detección de Huellas para la Conservación de Tortugas.....	16
3. Estado del Arte	18
3.1. Contexto de la Detección Automatizada en la Conservación Marina	18
3.2. Avances Recientes en la Detección Automatizada	19
3.2.1 Avances tecnológicos en componentes físicos de computadoras.	19
3.2.2 Progresos en el desarrollo de Algoritmos de Aprendizaje Profundo	19
3.2.3 Integración de Tecnologías	20
3.3. Uso de Deep Learning en la Monitorización Ambiental.....	21
3.3.1 Aplicaciones de la Monitorización Ambiental.....	21
Monitoreo de la deforestación y uso del suelo	22
Detección de contaminación del agua y la calidad del aire	22
Detección de especies en peligro y biodiversidad	23
Aplicaciones en agricultura y cambio climático	23
3.3.2 Eficiencia y Precisión	23
3.4. Procesamiento de Imágenes en Condiciones Compleja	24
3.4.1. Preprocesamiento de Imágenes.....	25
3.4.2. Mejora del Contraste	25
3.4.3. Segmentación de Huellas	26

3.4.4. Validación de Resultados.....	26
3.5. Comparativa de Algoritmos de Detección y Clasificación	26
3.5.1 YOLO (You Only Look Once).....	27
3.5.2 U-Net	28
3.5.3 R-CNN y Variantes.....	30
3.5.4 SSD (Single Shot Multibox Detector).....	31
3.5.5 Comparativa Directa	32
3.5.6 Aplicación en la Conservación de Tortugas Marinas.	33
3.6. Herramientas y Tecnologías Complementarias	34
3.6.1 Herramientas de Etiquetado de Imágenes	34
3.6.2 Tecnologías Adicionales	35
3.7. Validación y Evaluación de Modelos en Entornos Reales.....	36
3.7.1. Objetivos de la Validación	36
3.7.2. Selección del Entorno de Pruebas	36
3.7.3. Metodología de Pruebas	36
3.7.4. Evaluación de Resultados.....	37
4. Metodología	38
4.1. Recopilación y Tratamiento de Datos	38
4.2. Etiquetado de Imágenes.....	40
4.3. Configuración y Ajuste de los Modelos YOLO y Unet	43
4.3.1. Modelo YOLO	43
4.3.2. Modelo UNet.....	46
4.4. Validación y Pruebas del Modelo.....	51
5. Resultados.....	53
5.1. Explicación Dataset	53
5.2. Evaluación de la Precisión y Eficacia del Modelo	54
5.2.1. Resultados del Modelo YOLO	54
5.2.2. Resultados del Modelo UNet.....	57
5.3. Análisis de los Resultados	59
5.3.1 Desempeño del Modelo YOLOv8.....	59
5.3.2 El desempeño del Modelo U-Net.....	60
5.3.3 Evaluación comparada entre YOLOv8 y U-Net	62
5.4. Discusiones	62

5.4.1 Observación de las Fortalezas Identificadas:	62
5.4.2 Áreas de Mejora Identificadas	63
5.4.3 Relevancia en la Conservación de la Biodiversidad	63
6. Conclusiones y Trabajos Futuros	65
6.1. Conclusiones y Contribuciones	65
6.1.1. Resultados Relevantes en el Reconocimiento y Segmentación de Huellas de Tortuga	65
6.1.2. Impacto en el Campo de la Conservación Marina	65
6.1.3. Evaluación Integral de Modelos con Complementariedad Estratégica	66
6.1.4. Adaptación y Robustez ante Condiciones Desafiantes	66
6.1.5. Innovación en el Proceso de Creación de Conjuntos de Datos	66
6.1.6. Limitaciones y Áreas de Mejora	66
6.1.7. Impacto a Largo Plazo en la Sostenibilidad y Conservación	67
6.2. Trabajo Futuro	67
7. Bibliografía	68
8. Anexos	76
8.1. Script Principal	76
8.2. Script unir_dataset	76
8.3 Script Unir_Dividir_Manual	78
8.4 Script Porcentajes:	80

Lista de Figuras

Figura 1 Cronograma de actividades del Trabajo de Fin de Máster (Elaboración propia).....	11
Figura 2 Diagrama de Gantt del Proyecto TFM (Elaboración propia)	12
Figura 3 Proceso de monitoreo marino para la conservación de tortugas marinas (Creación propia por IA)	15
Figura 4 Ejemplos de capturas nocturnas y diurnas para el monitoreo de fauna en Doñana (Fuente: Fundación Descubre, 2016).....	16
Figura 5 Representación conceptual del monitoreo de huellas y tortugas marinas mediante inteligencia artificial y drones (Elaboración propia mediante inteligencia artificial).....	17
Figura 6 Imagen etiquetada proporcionada por DELT (Fuente: DELT)	20
Figura 7 Imagen etiquetada proporcionada por DELT (Fuente: DELT)	21
Figura 8 Arquitectura de una Red Neuronal Convolutacional (CNN) (Fuente: Towards Data Science, 2018)	22
Figura 9 Proceso de selección y entrenamiento de modelos (Fuente: O'Reilly, 2019)	23
Figura 10 Flujo de Procesamiento de Imágenes Nocturnas para la Detección de Huellas de Tortuga (Elaboración propia).....	24
Figura 11 Comparación de Imágenes de Resonancia Magnética: Original vs. Mejorada con SubtleMR (Fuente: Fuente: Foro Salud Digital, 2023)	25
Figura 12 Estructura del Modelo YOLO (Fuente: Liu et al., 2016)	27
Figura 13 Arquitectura U-Net (Fuente: datascientists, Daniel, 2023).....	29
Figura 14 Estructura de RCNN (Fuente: Ren et al., n.d.)	30
Figura 15 Estructura de SSD (Fuente: Liu et al., 2016)	31
Figura 16 Comparativa de las características principales de algoritmos de detección y segmentación de objetos (elaboración propia)	33
Figura 17 Fases del procesamiento del dataset (elaboración propia).....	38
Figura 18 Estructura del dataset proporcionado por DELT (elaboración propia).	40
Figura 19 Flujo de Trabajo de Imágenes con LabelMe (elaboración propia)	41
Figura 20 Ejemplo de Etiquetado de Huellas en Imagen Nocturna utilizando LabelMe (elaboración propia)	42
Figura 21 Etiquetado de Huellas en Imagen Diurna con Alta Densidad de Marcas (elaboración propia)	42
Figura 22 Esquema del Proceso de Implementación del Modelo YOLOv8 (elaboración propia)	44
Figura 23 Esquema del Funcionamiento del Modelo YOLO (elaboración propia).....	45
Figura 24 Ejemplo de imagen nocturna del dataset DELT y su correspondiente máscara generada para U-Net (elaboración propia).	47
Figura 25 U-Net funcionamiento (elaboración propia).	49
Figura 26 Esquema del proceso de implementación y evaluación de U-Net para la segmentación de huellas de tortugas (elaboración propia).....	50
Figura 27 Esquema del proceso de validación y evaluación de los modelos YOLOv8 y U-Net (elaboración propia).	52
Figura 28 Matrices de Confusión de YOLOv8	55
Figura 29 Curva Precision-Recall de YOLOv8	55

Figura 30 Curva F1-Confianza de YOLOv8.....	55
Figura 31 Ilustraciones de predicciones de YOLOv8.....	56
Figura 32 Ilustraciones de predicciones de YOLOv8 2.....	56
Figura 33 Matriz de confusión promedio de U-Net (elaboración propia).....	57
Figura 34 Curvas de pérdida y precisión durante el entrenamiento del modelo U-Net (elaboración propia)	58
Figura 35 Resultados del Modelo U-Net: Comparación entre Imagen Original, Máscara Real y Predicción (Elaboración Propia).....	58
Figura 36 Resultados del Modelo U-Net: Comparación entre Imagen Original, Máscara Real y Predicción (Elaboración Propia).....	58
Figura 37 Resultados de Evaluación del Modelo YOLO: Precisión, Recall, mAP y Desempeño en Condiciones Adversas (Elaboración Propia)	60
Figura 38 Resultados de Evaluación del Modelo U-Net: Segmentación Precisa y Desempeño en Condiciones Complejas (Elaboración Propia)	61

1.Introducción

1.1. Contexto y Justificación del Trabajo

El propósito de este trabajo es la creación de un sistema automatizado fundamentado en modelos avanzados de aprendizaje profundo, que nos ayude abordar la problemática de la identificación precisa y eficaz de las huellas de las tortugas marinas en su entorno natural, una labor esencial para su conservación que en la actualidad se encuentra sujeta a procesos manuales laboriosos y costosos en términos de tiempo y recursos humanos. La resolución de esta problemática permitirá la optimización del seguimiento de las tortugas marinas, incrementando la calidad y la disponibilidad de información sobre sus patrones migratorios, hábitats y estado de salud de la población. Esta información resulta fundamental para la elaboración de estrategias de conservación fundamentadas en información actualizada y fidedigna.

La preservación de las tortugas marinas se ve confrontada con diversos desafíos debido al impacto de actividades humanas tales como la pesca no sostenible, el turismo en gran escala y la contaminación, factores que han contribuido a la reducción de sus poblaciones. Estas especies juegan un papel crucial en la estabilidad de los ecosistemas marinos, especialmente en la salud de los pastos marinos y los arrecifes de coral, donde fomentan la biodiversidad y la estabilidad ecológica.

La vigilancia de las huellas y rastros de las tortugas, fundamental para comprender sus patrones migratorios, áreas de alimentación y comportamientos reproductivos, constituye un instrumento esencial para la elaboración de estrategias de conservación eficaces. No obstante, estos procedimientos en la actualidad se basan en métodos manuales que requieren un esfuerzo significativo en términos de tiempo y recursos humanos, lo cual limita la cantidad y calidad de los datos recolectados. Adicionalmente, la diversidad de los entornos naturales, tales como las fluctuaciones de la iluminación y las particularidades de los fondos, añade complejidad al proceso.

En el presente escenario, se presenta la necesidad de desarrollar sistemas automatizados que faciliten la gestión de estas restricciones. Esta investigación propone un enfoque innovador fundamentado en modelos avanzados de aprendizaje profundo, con el objetivo de automatizar la identificación y análisis de las huellas de tortugas en imágenes capturadas en su ambiente natural. La automatización no solo disminuiría los costos y el tiempo requerido para la recolección de datos, sino que también optimizaría la precisión y consistencia de los resultados, facilitando la generación de información más sólida y confiable para los programas de conservación.

Dentro de las tecnologías disponibles, los modelos de detección de objetos como YOLO (You Only Look Once) se distinguen por su habilidad para llevar a cabo análisis en tiempo real con alta precisión, incluso en circunstancias complejas. YOLO analiza la imagen completa en una única pasada, lo cual lo posiciona como un instrumento efectivo para entornos dinámicos y con condiciones de luz fluctuantes. En contraposición, técnicas complementarias como U-Net tienen la capacidad de realizar segmentaciones exhaustivas de las huellas, lo que facilita un análisis granular de las áreas ocupadas y permite la realización de investigaciones de mayor calidad cualitativa. La integración de estas tecnologías posibilita satisfacer tanto la demanda de detección rápida como la exactitud en el análisis, adaptándose a las características específicas de cada entorno.

Además de maximizar el esfuerzo en la recolección de datos, este sistema automatizado posee la capacidad de modificar el enfoque de los programas de conservación al proporcionar información en tiempo real y reducir la intervención humana directa. Esto facultará a los especialistas en conservación para concentrar sus esfuerzos en tareas estratégicas, tales como la valoración de los impactos humanos, la salvaguarda de áreas críticas y la puesta en marcha de medidas proactivas para atenuar las amenazas a las que se enfrentan estas especies.

La conservación de las tortugas marinas es crucial para preservar el equilibrio ecológico de los ecosistemas marinos, ya que desempeñan un papel fundamental en su salud. Algunas especies, las cuales se encuentran en riesgo de extinción, juegan un papel fundamental en los ecosistemas marinos, como los pastos marinos y los arrecifes de coral, al desempeñar funciones específicas que promueven la biodiversidad. No obstante, la actividad humana ha tenido un impacto negativo considerable en las poblaciones de estos seres marinos, debido a actividades como la pesca, el turismo y la contaminación.

En la actualidad, en el ámbito de la conservación, se asignan considerables recursos humanos y temporales a la tarea de llevar a cabo la vigilancia manual de las huellas y rastros de tortugas. Esto con el fin de recopilar información valiosa acerca de sus patrones migratorios, zonas de alimentación y conducta reproductiva. Gracias a esta información, los conservacionistas y biólogos pueden diseñar áreas de protección de manera efectiva y analizar el impacto de las actividades humanas en las poblaciones de tortugas.

Surge como una solución innovadora para optimizar los esfuerzos el desarrollo de una herramienta automatizada de detección de huellas de tortugas basada en deep learning. Al proponer la implementación de un modelo de aprendizaje profundo para la identificación de huellas de tortugas en su hábitat natural, este proyecto busca agilizar la recopilación de datos con precisión y eficiencia. Esto contribuirá a optimizar la toma de decisiones en las estrategias de conservación. La automatización de dicho procedimiento puede disminuir tanto el tiempo como los gastos relacionados, lo que posibilita que el personal se enfoque en actividades más significativas en el ámbito de los programas de conservación.

1.2. Motivación Personal

Mi compromiso con la preservación del ecosistema marino se deriva de mi profunda vinculación con el mar, una relación que he desarrollado y robustecido a lo largo de los años gracias a mi experiencia como patrón de yate. Este vínculo personal con los océanos no solo me ha brindado la oportunidad de valorar su riqueza inmensa, sino también de tomar conciencia de las amenazas a las que se enfrentan sus ecosistemas, en particular las especies vulnerables como las tortugas marinas.

Por otro lado, mi entusiasmo por la tecnología, particularmente por las aplicaciones del aprendizaje profundo (deep learning), se encuentra en consonancia con mi anhelo de contribuir de manera activa a la conservación del medio marino. Esta iniciativa constituye una oportunidad excepcional para combinar mis competencias técnicas con mi interés en la conservación, mediante el desarrollo de herramientas innovadoras que no solo fomenten el conocimiento científico, sino que también optimicen los esfuerzos de protección de especies marinas en peligro.

La implementación de técnicas avanzadas de análisis de imágenes para identificar huellas de tortugas no solo representa un reto técnico emocionante, sino también un método tangible para contribuir a la conservación de estos animales emblemáticos. Las tortugas marinas constituyen indicadores cruciales de la condición de los ecosistemas y su preservación resulta indispensable para preservar el equilibrio ecológico. Mediante la realización de este proyecto, mi experiencia en navegación se transforma en un recurso valioso para entender las peculiaridades del entorno marino y guiar el desarrollo de soluciones tecnológicas.

Mi propósito último es aportar a la edificación de un vínculo entre la tecnología y la conservación, evidenciando cómo el aprendizaje profundo puede ser un instrumento potente para la resolución de problemas críticos en el campo de la ecociencia. Esta iniciativa tiene como objetivo no solo la protección de las tortugas marinas, sino también la inspiración de innovadoras formas de emplear la inteligencia artificial en la conservación de la biodiversidad.

1.3. Objetivos del Trabajo

Objetivo principal:

- Creación de un sistema automatizado para la identificación de rastreos de huellas de tortugas marinas en imágenes capturadas en su entorno natural, asegurando su exactitud, eficacia y capacidad de adaptación a diversas condiciones medioambientales.

Objetivos secundarios

- Elaborar un conjunto de datos de entrenamiento apropiado a través de la anotación manual meticulosa de imágenes que presentan huellas de tortugas, garantizando la calidad y diversidad requeridas para el adiestramiento del modelo.
- Optimizar y ajustar modelos de aprendizaje profundo, tales como YOLO, con el fin de incrementar la eficiencia en la detección de huellas, teniendo en cuenta las variaciones en las condiciones de iluminación, el tipo de sustrato y otros factores ambientales.
- Corroborar la efectividad y la solidez del sistema propuesto a través de pruebas rigurosas en diversos conjuntos de datos, tanto en entornos controlados como en escenarios reales, evaluando su desempeño con métricas estándar tales como precisión, recall y F1-score.
- Diseñar una herramienta práctica e intuitiva que promueva la adopción del sistema por parte de conservacionistas y biólogos, fomentando su incorporación en programas de vigilancia y conservación de tortugas marinas.

1.4. Descripción de la metodología utilizada en el desarrollo del proyecto

La metodología sugerida para este proyecto se estructura en cuatro etapas fundamentales, concebidas para asegurar la eficacia y la aplicabilidad del sistema de detección automatizada de huellas de tortugas marinas. Estas fases se enfocan en la recolección y preparación de información, la configuración y ajuste del modelo, el adiestramiento del sistema y la validación final para evaluar su rendimiento. Se procederá a detallar cada una de las fases.

- Fase 1: Recopilación y Etiquetado de Datos

La primera fase se centrará en la adquisición de un conjunto de imágenes representativas de las huellas de las tortugas en su ambiente natural. Se procederá a la recolección de estas imágenes a través de colaboraciones con programas de conservación, investigaciones de campo y fuentes de datos preexistentes. Considerando que la calidad del dataset es esencial para el éxito del modelo, se implementará un proceso de anotación manual riguroso con el fin de identificar las huellas presentes en las imágenes.

Para la realización de esta tarea, se emplearán herramientas de etiquetado como LabelImg o LabelMe, las cuales facilitan la delimitación de las huellas mediante cuadros delimitadores, especificando con exactitud su localización y dimensiones. Además, se

procurará asegurar la diversidad en el compendio, incluyendo imágenes capturadas bajo diversas condiciones de iluminación y variedades de sustrato, con el propósito de potenciar la habilidad del modelo para generalizar en variados contextos ambientales.

- Fase 2: Configuración y Ajuste del Modelo

A lo largo de esta etapa, se seleccionará una arquitectura de aprendizaje profundo apropiada para la tarea, evaluando alternativas que puedan mantener un equilibrio entre velocidad y precisión. Se llevarán a cabo métodos de preprocesamiento con el fin de optimizar los datos de entrada, tales como la mejora del contraste y la normalización de imágenes.

La modificación de parámetros se llevará a cabo de manera iterativa, empleando conjuntos de validación para ajustar hiperparámetros tales como la tasa de aprendizaje y el tamaño de los lotes. La meta consiste en construir un modelo sólido que pueda identificar huellas en diversas condiciones ambientales, minimizando problemas como el sobreajuste.

- Fase 3: Entrenamiento del Modelo

El modelo será entrenado empleando las imágenes previamente etiquetadas. Este procedimiento comprenderá:

- Alimentar al sistema con imágenes y etiquetas para que aprenda a identificar y localizar huellas de manera independiente.
- Emplear estrategias de incremento de datos (tales como rotaciones, modificaciones en la iluminación, etc.) con el fin de fortalecer el modelo ante las fluctuaciones en el entorno.
- Evaluar su rendimiento durante el entrenamiento a través de indicadores intermedios que faciliten la identificación y rectificación de posibles insuficiencias.

El objetivo de esta etapa es garantizar que el sistema adquiera eficazmente las características distintivas de las huellas en contextos complejos.

- Fase 4: Validación

Después de completar el entrenamiento del modelo, se llevarán a cabo la validación de será meticulosa y se llevará a cabo empleando un conjunto de datos autónomo que no haya sido observado por el modelo durante el entrenamiento. Se llevará a cabo una evaluación del rendimiento del sistema a través de métricas tales como:

- Precisión: Exactitud en la detección de huellas correctas.

- Recall: Capacidad para identificar todas las huellas presentes.
- F1-score: Balance entre precisión y recall.
- Robustez: Rendimiento en condiciones adversas, como baja iluminación o fondos irregulares.

Los hallazgos facilitarán la evaluación de la aplicabilidad del sistema en situaciones reales y su potencial incorporación en programas de conservación.

1.5. Planificación del Trabajo

Actividad	Fecha de inicio	Fecha de entrega
Definición del TFM: enunciado y entrega	01/10/2024	13/10/2024
Estado del arte: enunciado y entrega de la actividad	14/10/2024	03/11/2024
Solicitud de imágenes al tutor	25/10/2024	29/10/2024
Etiquetado de imágenes	30/10/2024	09/11/2024
Implementación: enunciado y entrega de la actividad	11/11/2024	15/12/2024
Generación del modelo	20/11/2024	30/11/2024
Pruebas iniciales del modelo	01/12/2024	09/12/2024
Evaluación de resultados	10/12/2024	14/12/2024
Ajustes del modelo	15/12/2024	17/12/2024
Comprobaciones finales	18/12/2024	22/12/2024
Redacción de la memoria: entrega preliminar	16/12/2024	22/12/2024
Redacción de la memoria: entrega final	23/12/2024	29/12/2024
Presentación audiovisual del trabajo	30/12/2024	07/01/2025
Entrega de la documentación al tribunal	08/01/2025	10/01/2025
Defensa pública del trabajo	13/01/2025	31/01/2025

Figura 1 Cronograma de actividades del Trabajo de Fin de Máster (Elaboración propia)

Diagrama de Gantt:

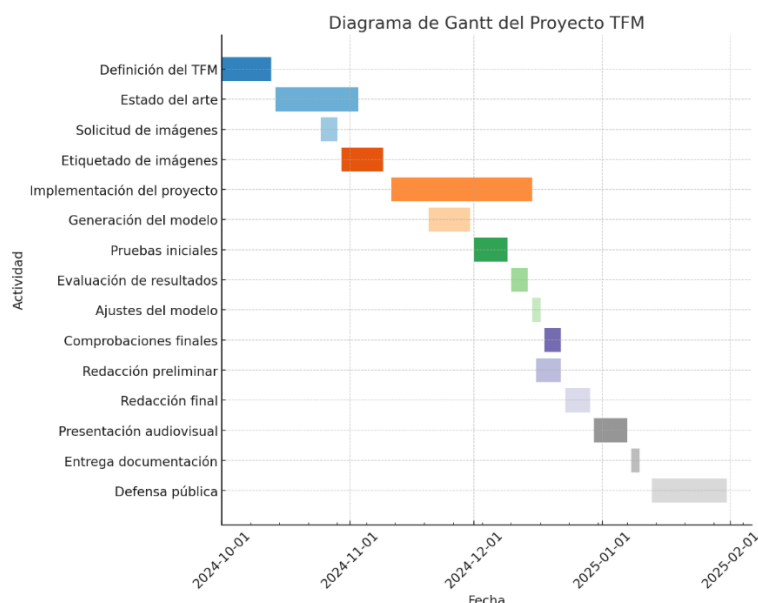


Figura 2 Diagrama de Gantt del Proyecto TFM (Elaboración propia)

1.6. Impacto en Sostenibilidad, Ético-social y de Diversidad

La tecnología desempeña un papel crucial en el progreso de soluciones sostenibles y en la reducción de problemas ambientales a nivel mundial. En este contexto, el proyecto se enfoca en la conservación marina, concretamente en la preservación de especies en riesgo como las tortugas marinas. Esta iniciativa genera efectos positivos en múltiples aspectos, incluyendo la sostenibilidad, la ética social y la diversidad.

1.6.1 Impacto en Sostenibilidad

La conservación de las tortugas marinas es fundamental para mantener la sostenibilidad de los ecosistemas marinos, dado que estas especies cumplen funciones esenciales en el equilibrio ecológico. Según Bjorndal (1997), las tortugas verdes desempeñan un papel crucial en la preservación de la salud de los pastos marinos, lo cual resulta beneficioso para la diversidad de especies presentes en el ecosistema y contribuye a la estabilidad del ciclo de nutrientes. La utilización de tecnologías avanzadas de detección automatizada para monitorear las huellas de tortugas marinas posibilita una intervención más precisa y efectiva, lo que conlleva a una disminución del impacto humano directo en sus hábitats (Spotila, 2004). Al favorecer la optimización en la identificación de rastros de tortugas, este proyecto reduce la dependencia de intervenciones manuales y promueve la implementación de prácticas sostenibles en la preservación de la biodiversidad marina.

1.6.2 Impacto Ético-Social

El impacto ético-social es un aspecto fundamental a considerar en cualquier estudio o análisis. Es crucial evaluar las implicaciones éticas y sociales de nuestras acciones y decisiones, ya que estas pueden tener efectos significativos en la sociedad en su conjunto. Es necesario reflexionar sobre cómo nuestras elecciones pueden afectar a la conservación de la fauna marina es considerada una responsabilidad compartida por las sociedades contemporáneas desde una perspectiva ética. Se destaca la importancia de mitigar los impactos negativos generados por actividades humanas, como la contaminación y la pesca no sostenible, según señala Mathews (2016). El proyecto promueve una relación de respeto con el entorno natural, favoreciendo el bienestar de especies en situación de vulnerabilidad. La automatización en la monitorización de las huellas, además, posibilita que los conservacionistas y voluntarios centren sus recursos en intervenciones más críticas, lo que contribuye a mejorar la eficiencia de las acciones de conservación. En definitiva, el proyecto establece una concordancia entre los objetivos de conservación y los principios éticos que promueven la convivencia armoniosa con el medio ambiente.

1.6.3 Impacto en Diversidad

Impacto en la diversidad es un tema de gran relevancia en la sociedad actual. La diversidad se refiere a la variedad de características que existen entre individuos y grupos, como la raza, el género, la orientación sexual, la edad, la discapacidad, entre otros.

La diversidad biológica juega un papel fundamental en la capacidad de recuperación de los ecosistemas, así como en la garantía de la seguridad alimentaria y económica de las comunidades locales que tienen una dependencia directa de los mismos (Chapin et al., 2000). El presente proyecto respalda la conservación de la diversidad biológica, la cual conlleva beneficios tanto ambientales como socioeconómicos. Según Wilson y Tisdell (2003), el turismo ecológico relacionado con la vida marina puede representar una fuente de ingresos para las comunidades costeras. La iniciativa promueve el respeto hacia la diversidad de especies presentes en los ecosistemas, concienciando a la sociedad acerca de la relevancia de cada organismo en el ciclo de vida marina.

En resumen, la incorporación de herramientas tecnológicas avanzadas en la vigilancia ambiental favorece una conservación más efectiva y ética, fomentando prácticas sostenibles y respetuosas con la biodiversidad y el equilibrio natural. Los esfuerzos mencionados tienen una relevancia que trasciende la preservación de especies particulares, ya que impactan directamente en la sostenibilidad a nivel mundial y en las interacciones futuras entre los seres humanos y el entorno natural (Naess, 1989).

2. Marco Teórico

2.1. Técnicas de Visión por Computadora para la Conservación

La visión por computadora ha generado una revolución en el ámbito de la conservación ambiental, ofreciendo instrumentos capaces de automatizar tareas complejas tales como la identificación, clasificación y seguimiento de especies en variados contextos naturales. Estas metodologías facilitan la obtención de datos pertinentes a partir de imágenes y vídeos, proporcionando soluciones más ágiles y exactas en comparación con los procedimientos manuales convencionales.

Las aplicaciones fundamentales de la visión computarizada en el campo de la conservación comprenden la identificación de especies, la detección de huellas, el análisis de patrones y la monitorización de hábitats. Estas metodologías resultan fundamentales para el monitoreo de especies en peligro de extinción en ecosistemas perturbados por la intervención humana. Instrumentos tales como cámaras trampa y vehículos aéreos no tripulados equipados con visión por computadora han facilitado la adquisición de datos de alta resolución en tiempo real, abarcando zonas extensas y de acceso complicado. De acuerdo con Goodwin et al. (2021), estos sistemas han disminuido la necesidad de esfuerzo manual, optimizando la frecuencia y calidad de las observaciones de especies como las tortugas y los tiburones.

Adicionalmente, su habilidad para funcionar en situaciones de baja iluminación o fondos de alta complejidad las convierte en particularmente valiosas en el ámbito de la conservación marina. Los algoritmos sofisticados de detección de bordes y segmentación han perfeccionado la detección de huellas en playas de anidación, un paso fundamental para el monitoreo de especies en peligro de extinción como las tortugas marinas.

Las metodologías computacionales proporcionan un enfoque sostenible y escalable para la conservación, minimizando la intervención humana y fomentando estrategias no invasivas, evidenciando su papel fundamental en la protección de la biodiversidad y los ecosistemas.

La siguiente figura ilustra el proceso laboral en la aplicación de visión por computadora destinada a la conservación de especies marinas. El proceso se inicia con la recopilación de imágenes a través de drones y cámaras trampa en playas y océanos, lo que facilita la supervisión de áreas distantes. Estas representaciones visuales son procesadas mediante el uso de inteligencia artificial, empleando algoritmos como las Redes Neuronales Convolucionales (CNN) para la identificación y segmentación. En última instancia, los datos se exponen de manera accesible para los conservacionistas, lo que simplifica la

detección de huellas, el monitoreo de patrones y la formulación de decisiones fundamentadas en datos contemporáneos.

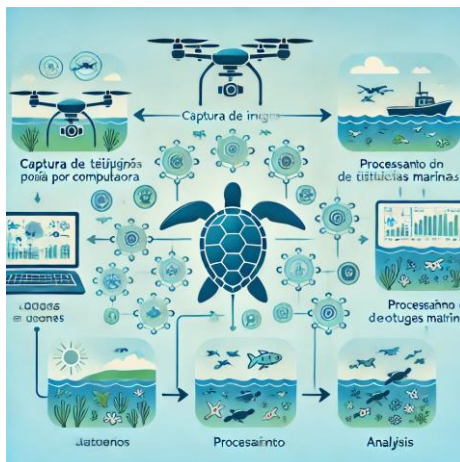


Figura 3 Proceso de monitoreo marino para la conservación de tortugas marinas (Creación propia por IA)

2.2. Aprendizaje Profundo en la Conservación

El aprendizaje profundo, también conocido como deep learning, ha transformado la manera en que los investigadores abordan los problemas de conservación de biodiversidad. Estas técnicas, basadas en redes neuronales profundas, permiten analizar grandes volúmenes de datos ambientales de forma más eficiente y precisa que los métodos tradicionales. Desde la identificación automática de especies hasta el monitoreo de cambios en ecosistemas, el aprendizaje profundo ha demostrado ser una herramienta clave para la sostenibilidad ambiental.

Un ejemplo sobresaliente es la implementación de algoritmos de aprendizaje profundo en cámaras de fototrampeo para la identificación automática de fauna. Por ejemplo, en el Parque Nacional de Doñana, los académicos han desarrollado sistemas que permiten la clasificación automática de las especies registradas, lo que ha disminuido considerablemente el esfuerzo manual y ha mejorado la exactitud de los censos de fauna. El ejemplo ilustrativo de estas capturas incluye imágenes tomadas en diversas condiciones ambientales y de iluminación, tales como imágenes diurnas, nocturnas y con niveles variables de contraste. Estas cámaras posibilitan la supervisión constante y no intrusiva de zonas protegidas, optimizando la recolección de datos cruciales para la conservación.

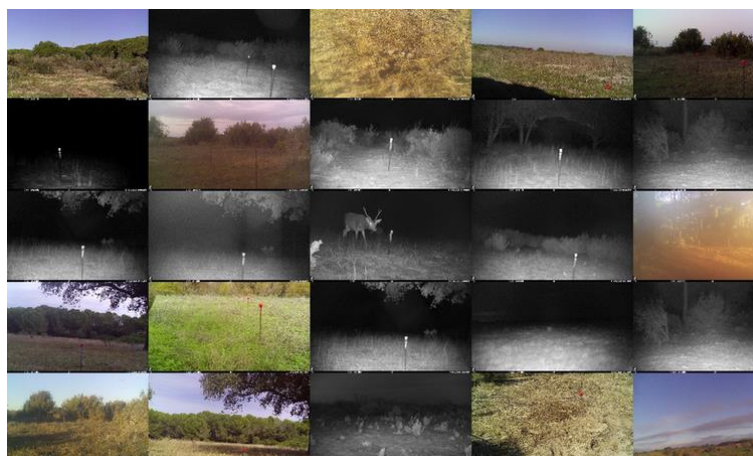


Figura 4 Ejemplos de capturas nocturnas y diurnas para el monitoreo de fauna en Doñana (Fuente: Fundación Descubre, 2016)

Proyectos como AI-CENSUS han instaurado sistemas automáticos fundamentados en el aprendizaje profundo para la supervisión de la biodiversidad, con un enfoque predominante en los mamíferos, aunque con potenciales aplicaciones para otros organismos. Estas herramientas maximizan la eficiencia de los recursos y facilitan la supervisión a gran escala, un aspecto esencial para la preservación en contextos naturales de alta complejidad.

Adicionalmente, el aprendizaje profundo no se circunscribe únicamente a la identificación de especies, sino que también posibilita el seguimiento de cambios en hábitats naturales, identifica actividades ilícitas como la caza ilegal y anticipa las repercusiones ambientales derivadas del cambio climático. Estas habilidades confieren al aprendizaje profundo un papel crucial en la protección de especies amenazadas y la mitigación de las amenazas a la biodiversidad.

La utilización de imágenes registradas bajo diversas circunstancias, tal como en el caso de Doñana, evidencia la habilidad de los sistemas automatizados para adaptarse a contextos complejos. Estas tecnologías posibilitan la identificación de especies, el análisis de patrones comportamentales y la identificación de alteraciones en los hábitats, lo que contribuye al fortalecimiento de estrategias de conservación en tiempo real. La integración de instrumentos de vanguardia y metodologías sólidas garantiza una salvaguarda sostenible y eficiente de los ecosistemas más susceptibles.

2.3. Importancia de la Detección de Huellas para la Conservación de Tortugas

La identificación de marcas de las tortugas marinas desempeña una función esencial en la preservación de dichas especies, dado que facilita la adquisición de datos detallados sobre su comportamiento y ecología. A través del análisis de las huellas, los investigadores

tienen la capacidad de calcular la cantidad de hembras anidantes, identificar zonas críticas de anidación y evaluar las tendencias poblacionales a través del tiempo. Este método no invasivo es crucial para el seguimiento de las tortugas marinas, muchas de las cuales están clasificadas como especies en peligro de extinción de acuerdo con la clasificación de la Unión Internacional para la Conservación de la Naturaleza (UICN).



Figura 5 Representación conceptual del monitoreo de huellas y tortugas marinas mediante inteligencia artificial y drones (Elaboración propia mediante inteligencia artificial)

Por ejemplo, en Guatemala, la playa de Hawaii ha emergido como un punto de referencia para el seguimiento de huellas de las tortugas parlama (*Lepidochelys olivacea*). Durante el periodo comprendido entre 2003 y 2019, se registró un aumento en la densidad de anidación, desde 906 huellas registradas en 2003 hasta 1,422 en 2019, lo que evidencia una tendencia poblacional positiva atribuible a los esfuerzos de conservación locales (ARCAS, 2019). Este tipo de seguimiento facilita a los investigadores no solo la comprensión de las dinámicas poblacionales, sino también la identificación de las zonas que demandan una mayor salvaguarda debido a su relevancia ecológica.

La detección de huellas promueve la salvaguarda de nidos, incrementando las tasas de eclosión, y posibilita el seguimiento de amenazas como predadores o intervenciones humanas, proporcionando información esencial para la formulación de estrategias de mitigación. Tecnologías avanzadas, tales como la visión por computadora y el aprendizaje profundo, optimizan la exactitud y eficiencia del monitoreo mediante la automatización de la detección y análisis de huellas en playas infestadas.

El mantenimiento de la integridad de los ecosistemas marinos es fundamental para la conservación de las tortugas marinas. Las tortugas verdes mantienen la integridad de los pastos marinos, mientras que las carey regulan las poblaciones de esponjas en los arrecifes marinos. La salvaguarda de estas especies garantiza igualmente la provisión de servicios ecosistémicos indispensables para las comunidades humanas. Por lo tanto, la identificación de huellas no se limita a ser un instrumento científico, sino que constituye un componente esencial en las estrategias globales de conservación.

3. Estado del Arte

3.1. Contexto de la Detección Automatizada en la Conservación Marina

La conservación de los ecosistemas marinos se ve confrontada por desafíos de gran magnitud, tales como la contaminación, el cambio climático y las prácticas de pesca no sostenibles. Estos factores inciden directamente en los ecosistemas y la biodiversidad marina, como señala Duarte (2020). Las tecnologías avanzadas de detección automatizada se han convertido en herramientas fundamentales para la vigilancia y preservación de la biodiversidad marina. Los sistemas mencionados posibilitan una recolección eficiente y precisa de datos, lo que simplifica el estudio de patrones y conductas en poblaciones marinas. Este tipo de análisis sería imposible de realizar mediante métodos manuales, según Long y colaboradores (2017). La incorporación de técnicas de visión por computadora y aprendizaje profundo en el ámbito de la conservación marina ha generado nuevas posibilidades para potenciar la efectividad y la amplitud de las iniciativas de preservación (Kachelriess et al., 2014).

Según Goodwin et al. (2021), se ha comprobado la eficacia de los métodos automatizados de detección, tales como la utilización de cámaras submarinas y drones aéreos, en la vigilancia de las poblaciones de especies marinas y sus respectivos entornos. Estas tecnologías posibilitan la realización de evaluaciones en tiempo real y brindan una amplia cobertura de áreas de difícil acceso, generando datos de alta calidad con riesgos reducidos tanto para los investigadores como para el entorno natural. La automatización, según Schofield et al. (2019), disminuye la labor manual en la identificación y conteo de ejemplares de especies en riesgo, como tortugas, tiburones y ballenas, mejorando la exactitud y frecuencia de los estudios de poblaciones marinas.

La detección y clasificación de imágenes de fauna marina mediante algoritmos de aprendizaje profundo se destaca por su potencial, ya que estos algoritmos pueden reconocer patrones complejos en imágenes y vídeos, incluso en entornos naturales con variaciones en la iluminación y fondos visuales complicados (Norouzzadeh et al., 2018). Los algoritmos, como YOLO (You Only Look Once), son ampliamente utilizados en aplicaciones de detección en tiempo real. Han sido modificados para identificar especies y sus rastros en el ámbito de la conservación. Esto posibilita una respuesta rápida ante la detección de amenazas o variaciones en los comportamientos de las especies (Redmon & Farhadi, 2018).

Los sistemas automatizados de detección, además de ser eficientes y rápidos, posibilitan la disminución de la influencia humana en los ecosistemas marinos. La utilización de técnicas no invasivas en la obtención de información puede reducir la ansiedad en los

organismos y en su hábitat, fomentando en consecuencia enfoques de conservación más sustentables (Hammond et al., 2019). La importancia de este aspecto se destaca en el seguimiento de especies en situación de vulnerabilidad, ya que es fundamental minimizar al máximo las acciones humanas con el fin de prevenir posibles consecuencias negativas.

La automatización de la detección en el ámbito de la conservación marina constituye un progreso fundamental, al fusionar precisión y sostenibilidad, lo que transforma la vigilancia de ecosistemas y optimiza la protección de la biodiversidad.

3.2. Avances Recientes en la Detección Automatizada

Los avances tecnológicos en la detección automatizada han tenido un impacto revolucionario en diversos sectores, entre ellos la conservación de la biodiversidad. En los últimos años, se ha observado un notable avance en la capacidad de detectar y monitorear de forma automática especies y sus rastros en entornos naturales. Este progreso se ha logrado gracias al desarrollo de nuevas técnicas de visión por computadora y aprendizaje profundo.

3.2.1 Avances tecnológicos en componentes físicos de computadoras.

Los avances tecnológicos recientes en hardware, tales como cámaras de alta resolución y sensores especializados, han posibilitado la captura de imágenes detalladas en diversas condiciones ambientales. La utilización de drones equipados con cámaras térmicas y de visión nocturna ha posibilitado la realización de investigaciones sobre poblaciones animales en su entorno natural sin perturbaciones humanas, generando información constante que resultaba complicada de adquirir en el pasado (Kays et al., 2020).

3.2.2 Progresos en el desarrollo de Algoritmos de Aprendizaje Profundo

Los algoritmos más sofisticados han contribuido significativamente a las mejoras observadas en el software de detección. Los algoritmos de aprendizaje profundo, tales como las Redes Neuronales Convolucionales (CNN) y YOLO (You Only Look Once), han sido mejorados con el fin de identificar y clasificar objetos con una precisión y velocidad sin precedentes. En la actualidad, es posible entrenar estos modelos utilizando grandes conjuntos de datos con el fin de identificar patrones específicos con alta precisión. Este aspecto resulta fundamental en el proceso de identificación de especies en imágenes obtenidas en entornos naturales (He et al., 2016).

La integración de tecnologías es un proceso fundamental en el ámbito académico y profesional, que implica la incorporación efectiva de diferentes herramientas tecnológicas en diversos contextos. Este proceso requiere una planificación cuidadosa y una comprensión profunda de las necesidades y objetivos específicos de cada situación.

3.2.3 Integración de Tecnologías

La capacidad de los investigadores para monitorear y gestionar datos sobre especies y sus hábitats se ha mejorado gracias a la integración de tecnologías de detección automatizada con sistemas de información geográfica (GIS) y bases de datos en línea. Las plataformas eBird y iNaturalist hacen uso de estos avances tecnológicos para posibilitar la participación de ciudadanos científicos e investigadores en la recopilación y consulta de datos sobre la biodiversidad a nivel mundial (Sullivan et al., 2014).

Los avances mencionados han mejorado la eficacia de los proyectos de conservación y han promovido la participación ciudadana en la investigación científica de la conservación. Esto ha permitido un mayor acceso a la información y ha contribuido al aumento de la conciencia ambiental.

La implementación de conjuntos de datos especializados, como por ejemplo el dataset "Arribada2023", está teniendo un papel fundamental en el avance de las tecnologías de detección nocturna. El conjunto de datos en cuestión contiene imágenes obtenidas en entornos con poca iluminación, lo que posibilita a los investigadores la tarea de analizar y categorizar diversos tipos de indicios de presencia de vida marina, como por ejemplo, las huellas dejadas por tortugas marinas.

La investigación de esta información ofrece información importante sobre las fluctuaciones en la visibilidad y el contraste de los rastros en función de las condiciones ambientales. Esto, a su vez, contribuye a la creación de modelos de detección más sólidos y flexibles. Los avances mencionados son esenciales para hacer frente a los retos que surgen en los ecosistemas nocturnos, cuando las actividades humanas han finalizado y numerosas especies marinas están en plena actividad. Esto demanda el desarrollo de tecnologías capaces de funcionar de manera eficiente en tales condiciones.

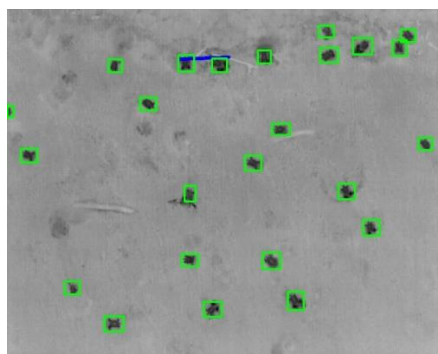


Figura 6 Imagen etiquetada proporcionada por DELT (Fuente: DELT)

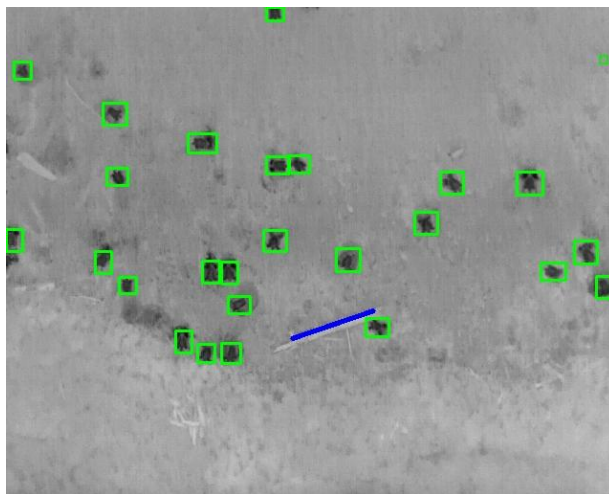


Figura 7 Imagen etiquetada proporcionada por DELT (Fuente: DELT)

La colaboración con organizaciones como DELT, que aplica técnicas avanzadas de multi-forma para etiquetar imágenes, complementa los esfuerzos mencionados. Esta colaboración integra nuevas dimensiones en el análisis y contribuye a mejorar la precisión de las herramientas automatizadas utilizadas en la monitorización.

3.3. Uso de Deep Learning en la Monitorización Ambiental

La aplicación de técnicas de aprendizaje profundo en la vigilancia del entorno ha revolucionado la capacidad de los investigadores para analizar y comprender extensos conjuntos de datos ambientales. La tecnología mencionada posibilita la creación de modelos predictivos y de clasificación más precisos y eficientes, los cuales son fundamentales para llevar a cabo evaluaciones en tiempo real de las condiciones ambientales y la biodiversidad.

3.3.1 Aplicaciones de la Monitorización Ambiental.

Las Redes Neuronales Convolucionales (CNN) son modelos de deep learning que se emplean extensamente en la clasificación de imágenes y la detección de cambios en el paisaje. Esta aplicación es fundamental para la identificación de zonas impactadas por fenómenos ambientales como la deforestación o la contaminación del agua (Zhu, X. et al., 2017). Podemos decir que las Redes Neuronales Convolucionales (CNN) se han consolidado como uno de los instrumentos más eficaces para enfrentar retos asociados con la supervisión ambiental.

Estas redes demuestran una eficacia particular en la categorización de imágenes y la identificación de alteraciones en el paisaje, dos funciones fundamentales para la

evaluación de la salud de los ecosistemas y la detección de amenazas emergentes. Las CNN facilitan la identificación de patrones visuales en imágenes de satélite, de vehículos aéreos no tripulados y de cámaras terrestres, lo que permite una clasificación automática y eficaz de elementos fundamentales en el ambiente natural.

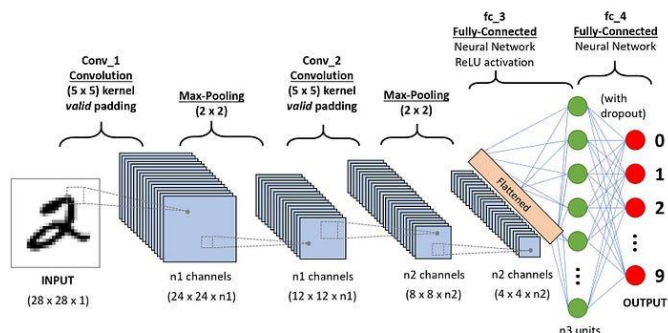


Figura 8 Arquitectura de una Red Neuronal Convolutiva (CNN) (Fuente: Towards Data Science, 2018)

Monitoreo de la deforestación y uso del suelo

Una de las disciplinas más afectadas por estas tecnologías es el seguimiento de la deforestación. La implementación de CNN en investigaciones recientes ha permitido el análisis de imágenes satelitales para identificar zonas afectadas por la tala ilegal de bosques y la expansión agrícola. Por ejemplo, Zhu et al. (2017) evidenciaron la capacidad de las CNN para identificar y categorizar áreas deforestadas con una precisión considerablemente superior a los métodos convencionales que se fundamentan en algoritmos estadísticos. Adicionalmente, la tecnología ha sido empleada para detectar alteraciones en la utilización del suelo en tiempo casi real, facilitando a los gobiernos y entidades no gubernamentales la toma de decisiones fundamentadas para la protección de ecosistemas en situación de vulnerabilidad.

Detección de contaminación del agua y la calidad del aire

Las CNN han evidenciado su eficacia en la identificación de la contaminación hídrica y en la supervisión de su calidad. Mediante el uso de imágenes multiespectrales y de satélite, los modelos son capaces de identificar indicadores de contaminación, tales como la proliferación de algas nocivas, los vertidos químicos y los sedimentos presentes en cuerpos de agua. Esta metodología ha desempeñado un papel crucial en la preservación de los recursos hídricos, dado que facilita una evaluación continua y no invasiva.

Además, el aprendizaje profundo ha facilitado la supervisión de la calidad atmosférica a través de imágenes satelitales y datos recopilados de sensores dispuestos en zonas urbanas e industriales. La integración de CNN con datos meteorológicos permite cartografiar las concentraciones de contaminantes y pronosticar su dispersión, proporcionando un instrumento esencial para la planificación ambiental y la mitigación de riesgos.

Detección de especies en peligro y biodiversidad

La detección de especies amenazadas constituye otra aplicación prominente de las CNN en la supervisión ambiental. Los algoritmos se han capacitado para categorizar de manera automática las imágenes capturadas por cámaras trampa en hábitats naturales. Por ejemplo, iniciativas como Wildlife Insights han utilizado la tecnología CNN para analizar millones de imágenes capturadas en reservas naturales, identificando especies objetivo y generando bases de datos que facilitan la planificación de iniciativas conservacionistas.

Aplicaciones en agricultura y cambio climático

Dentro del ámbito agrícola, las Ciencias de la Información (CNN) han sido empleadas para supervisar la salud de los cultivos e identificar plagas o enfermedades potencialmente perjudiciales para la productividad. Estas aplicaciones también se extienden al análisis de las repercusiones del cambio climático en los ecosistemas naturales.

3.3.2 Eficiencia y Precisión

La eficiencia y la precisión son aspectos fundamentales a considerar en cualquier proceso. La eficiencia se refiere a la capacidad de lograr un objetivo con la menor cantidad de recursos posibles, mientras que la precisión se relaciona con la exactitud y la calidad de los resultados obtenidos. Ambos conceptos son esenciales para garantizar la

El procesamiento de datos ambientales se beneficia de manera significativa de la eficiencia y automatización que ofrece el enfoque de deep learning, el cual supera a las técnicas convencionales en términos de velocidad y precisión. La importancia de este aspecto se destaca en aquellas aplicaciones en las que la precisión y la rapidez son fundamentales para la toma de decisiones y la respuesta a situaciones de emergencia ambiental.

Los avances en deep learning representan una herramienta de gran utilidad para las iniciativas de conservación y la administración sostenible del entorno, facilitando una supervisión más eficaz y fundamentada de los ecosistemas naturales.

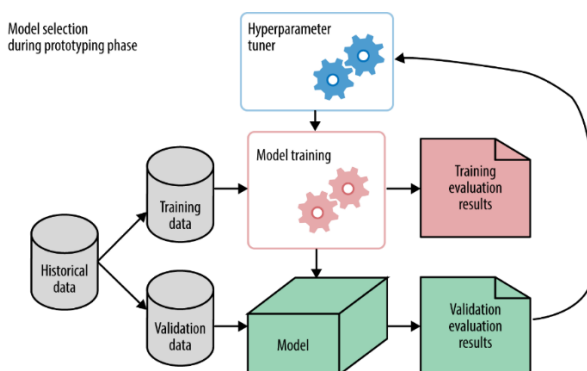


Figura 9 Proceso de selección y entrenamiento de modelos (Fuente: O'Reilly, 2019)

3.4. Procesamiento de Imágenes en Condiciones Compleja

El monitoreo nocturno es crucial para preservar especies como las tortugas marinas, ya que muchas de sus actividades, como la anidación, ocurren durante la noche (Schofield et al., 2019). Las imágenes nocturnas suelen tener bajo contraste y alto ruido, dificultando la detección automática de huellas. Para abordar esto, se han implementado técnicas de procesamiento de imágenes que optimizan el contraste y mejoran la detección en condiciones de poca luz.

Dentro del marco del proyecto de las imágenes obtenidas en 'Arribada2023' (proyecto del trabajo), se han examinado imágenes nocturnas con fluctuaciones en contraste y longitud de las huellas, empleando técnicas avanzadas de segmentación y aprendizaje profundo para evaluar su efectividad. Además, se han implementado filtros como Hough en OpenCV con el objetivo de extraer características y optimizar la precisión de los modelos, explorando metodologías innovadoras para el seguimiento de especies amenazadas.

La colaboración con el colaborador de DELT, quien está agregando anotaciones de multi-forma a algunas de nuestras imágenes, ha sido fundamental. El proceso de anotación detallada contribuye a la obtención de un conjunto de datos enriquecido que simplifica la evaluación y mejora constante de los modelos de detección, garantizando su capacidad de ajustarse a las variaciones propias de los entornos naturales nocturnos. Los resultados iniciales de estos exámenes se están recopilando para llevar a cabo una validación exhaustiva, cuyo objetivo es establecer criterios sólidos que garanticen la identificación eficaz de huellas en diversas condiciones de iluminación.

Flujo de Procesamiento de Imágenes Nocturnas para la Detección de Huellas de Tortuga:

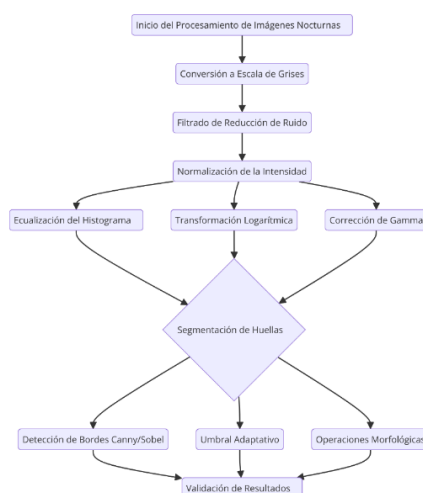


Figura 10 Flujo de Procesamiento de Imágenes Nocturnas para la Detección de Huellas de Tortuga (Elaboración propia)

3.4.1. Preprocesamiento de Imágenes

El preprocesamiento de imágenes es una etapa fundamental en el proceso de análisis, ya que permite disminuir la interferencia y acondicionar las imágenes para su posterior estudio. Se plantea el siguiente procedimiento de trabajo.

- La transformación a tonos de gris: El análisis se simplifica mediante la eliminación de información cromática superflua, lo cual disminuye el tiempo de procesamiento (Gonzalez y Woods, 2018).
- Reducción de ruido mediante filtrado: Se implementan filtros, tales como el Gaussiano o de mediana, con el objetivo de suavizar la imagen y preservar detalles cruciales (Jain, 1989).
- Normalización de la intensidad: Optimiza el contraste global y enfatiza las marcas en relación con el fondo oscuro, facilitando su identificación (Nixon & Aguado, 2020).

3.4.2. Mejora del Contraste

Una vez que las imágenes nocturnas han sido sometidas a un proceso de preprocesamiento, es necesario aplicar técnicas avanzadas de mejora de contraste para lograr la visualización de los rastros de tortugas en situaciones de escasa iluminación.

- Ecualización del histograma: Redistribuye de manera uniforme los niveles de intensidad con el objetivo de potenciar el contraste global y destacar las marcas en el lecho de arena (Pizer et al., 1987).
- Transformación logarítmica: Amplía la precisión en niveles de intensidad reducidos, optimizando la visibilidad en imágenes nocturnas (Oppenheim et al., 1999).
- Corrección de gamma: Adquiere tonalidades intermedias para destacar detalles sin impactar los tonos claros u oscuros (Petrou y Bosdogianni, 1999).

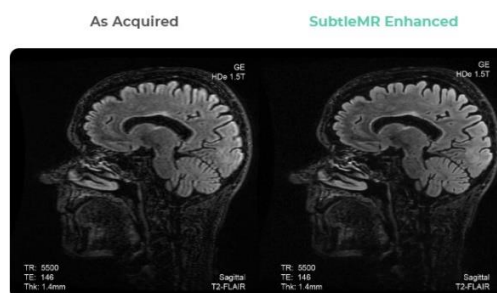


Figura 11 Comparación de Imágenes de Resonancia Magnética: Original vs. Mejorada con SubtleMR (Fuente: Foro Salud Digital, 2023)

3.4.3. Segmentación de Huellas

Para la extracción de las huellas de tortuga en las imágenes, se emplean técnicas de segmentación que permiten identificar los contornos y resaltar las zonas de interés.

La detección de bordes es un proceso fundamental en el campo del procesamiento de imágenes. Para identificar los contornos de las huellas en la arena, se recomienda utilizar detectores de bordes como el operador Canny o Sobel. Según Canny (1986), la detección de bordes es un proceso que posibilita la separación de la forma de las huellas del fondo, permitiendo así su diferenciación.

La segmentación basada en umbral adaptativo es un método utilizado en procesamiento de imágenes para dividir una imagen en regiones significativas. Este enfoque se basa en ajustar dinámicamente los umbrales de cada región de la imagen en función de sus características locales, lo que permite una segmentación más precisa y robusta en comparación con métodos estáticos de umbral fijo. Es necesario implementar un umbral adaptativo que pueda adaptarse a las fluctuaciones de iluminación presentes en la imagen. El presente enfoque resulta óptimo para imágenes que presentan variaciones de luz ocasionadas por la iluminación ambiente en entornos nocturnos (Bradley & Roth, 2007).

Operations of morphology: Para mejorar la forma y definir los contornos de las huellas, es recomendable aplicar operaciones morfológicas como la erosión y la dilatación. Estas operaciones ayudan a eliminar posibles restos de ruido en las imágenes, según lo indicado por Serra en 1982.

3.4.4. Validación de Resultados

Con el fin de garantizar la adecuada detección de huellas en el procesamiento y mejora de imágenes nocturnas, se procederá a evaluar el rendimiento del modelo mediante la utilización de métricas de precisión y recall. Este análisis se llevará a cabo a través de pruebas realizadas en imágenes que presenten distintas condiciones de iluminación reducida. La validación mencionada posibilita la adecuación de los parámetros y la optimización del modelo en diversos contextos, lo que contribuye a fortalecer la resistencia del sistema durante la noche.

3.5. Comparativa de Algoritmos de Detección y Clasificación

Como hemos observado la identificación automatizada y clasificación de objetos en imágenes ha experimentado una evolución notable gracias a los progresos en las redes neuronales profundas. Dentro del contexto de la conservación de la biodiversidad, estas

herramientas se erigen como fundamentales para la identificación de patrones, rastros y especies en contextos complejos, como las playas de anidación de tortugas acuáticas.

La finalidad primordial al elegir un algoritmo apropiado no solo es optimizar la precisión de las detecciones, sino también asegurar que el modelo pueda ajustarse a las condiciones particulares del entorno, tales como baja iluminación o fondos de alta complejidad. Adicionalmente, la eficiencia en cuanto a la velocidad y la capacidad de procesamiento en tiempo real se establece como un criterio esencial en tareas como la supervisión constante o la supervisión de extensas áreas.

Este apartado examina una variedad de algoritmos de visión computacional, subrayando sus beneficios, restricciones y aplicaciones. Métodos tales como U-Net proporcionan segmentación exacta, mientras que YOLO se enfoca en la detección acelerada en tiempo real. Además, se examinan métodos convencionales, como la R-CNN, para situaciones donde la precisión es esencial, evaluando su equilibrio en proyectos de conservación de tortugas marinas.

3.5.1 YOLO (You Only Look Once)

El enfoque YOLO (You Only Look Once) es una técnica utilizada en el campo de la visión por computadora para la detección de objetos en imágenes. El algoritmo YOLO destaca por su rapidez y eficacia en la detección de objetos en tiempo real. En contraste con los sistemas de detección que emplean el modelo en múltiples regiones de la imagen, YOLO aborda la detección como un problema de regresión única. Este enfoque consiste en realizar una predicción directa desde la imagen de entrada hasta las coordenadas del cuadro delimitador y las probabilidades de clase, todo ello mediante una sola red neuronal. Esto posibilita que el método YOLO logre la detección de objetos de manera considerablemente más rápida que enfoques como R-CNN y sus variantes (Redmon et al., 2016).

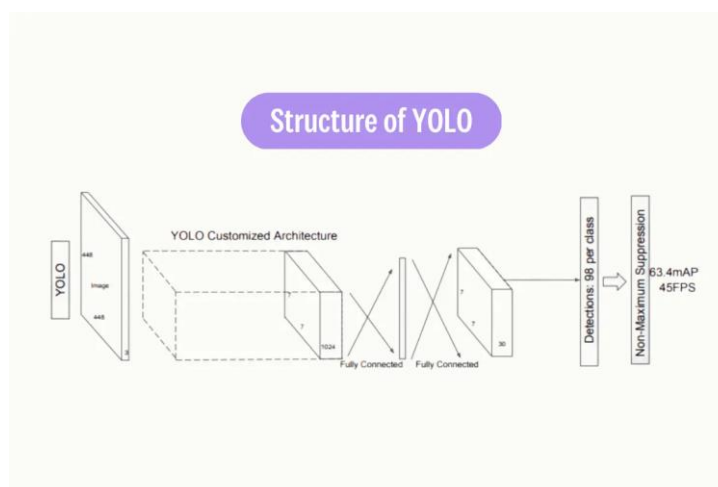


Figura 12 Estructura del Modelo YOLO (Fuente: Liu et al., 2016)

Ventajas :

- La velocidad es una magnitud física que expresa la relación entre el espacio recorrido por un objeto y el tiempo que ha tardado en recorrerlo. La capacidad de procesamiento en tiempo real de imágenes de YOLO es notablemente alta, lo cual resulta óptimo para situaciones donde la prontitud de la respuesta es fundamental, como en la vigilancia en tiempo real del entorno natural.
- Exactitud: Las mejoras en YOLOv3 y YOLOv4 han aumentado significativamente su precisión, volviéndolos competitivos con otros algoritmos avanzados en tareas de detección, a pesar de haber sido inicialmente menos precisos que métodos como Faster R-CNN (Redmon & Farhadi, 2018).
- La capacidad de generalización es un aspecto fundamental en la investigación científica. El detector YOLO presenta una menor tasa de errores de localización en contraste con otros detectores disponibles en el mercado. Durante el proceso de entrenamiento, el sistema adquiere representaciones generales de objetos, lo cual explica este fenómeno.

Limitaciones:

- Sensibilidad a la escala es un concepto fundamental en el análisis de datos. Las versiones originales de YOLO presentaban dificultades al detectar objetos de pequeño tamaño. No obstante, en versiones más recientes, esta situación ha sido parcialmente reducida gracias a la utilización de diversas escalas de predicción.
- Contexto del objeto: El algoritmo YOLO puede presentar dificultades en la identificación precisa de objetos cuando estos se encuentran en situaciones poco habituales, a pesar de que los avances en el proceso de entrenamiento y en la capacidad de la red han contribuido a mitigar esta problemática.

3.5.2 U-Net

El modelo U-Net constituye una arquitectura de redes neuronales convolucionales, específicamente concebida para tareas de segmentación semántica. La estructura simétrica de codificador-decodificador, complementada con conexiones de salto entre capas pertinentes, posibilita que U-Net combine características de alto nivel con detalles espaciales exactos, lo que la convierte en un instrumento particularmente valioso para la identificación de patrones en imágenes complejas (Ronneberger et al., 2015).

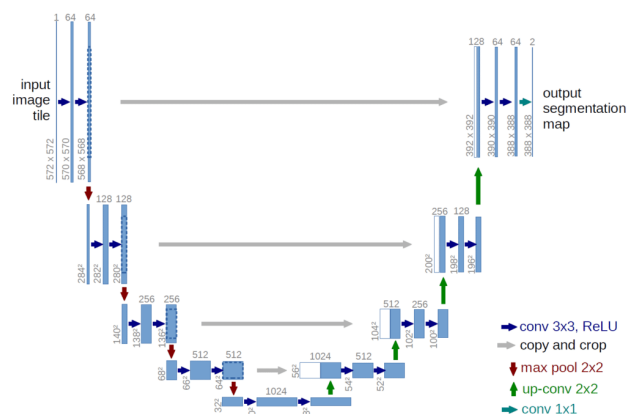


Figura 13 Arquitectura U-Net (Fuente: datascientists, Daniel, 2023)

Ventajas:

- **Precisión en la segmentación:** U-Net se distingue en tareas que demandan segmentación exacta, tales como la detección de formas irregulares y detalles pequeños, lo que resulta óptimo para la detección de huellas o rastros en imágenes con ruido o fondos complejos.
- **Adaptabilidad a la implementación:** La arquitectura de esta estructura es adaptable a una diversidad de problemas, desde representaciones médicas hasta contextos naturales, lo que la convierte en un instrumento versátil para la identificación de rastros de tortugas.
- **Eficiencia en contextos de datos restringidos:** La capacidad de U-Net para entrenar eficazmente conjuntos de datos relativamente pequeños se atribuye a su habilidad para optimizar el uso de los datos etiquetados mediante la combinación de características globales y locales.

Limitaciones:

- **Coste computacional:** El modelo U-Net puede ser más intensivo en términos de memoria y tiempo de procesamiento, particularmente en relación con otros métodos como YOLO.
- **Dependencia en la precisión de las anotaciones:** El desempeño de U-Net está considerablemente condicionado por la calidad de las etiquetas presentes en el conjunto de entrenamiento, lo que puede representar un obstáculo en situaciones con datos mal etiquetados o inconsistencias.

- Insuficiencia en la capacidad de tiempo real: Contrariamente a YOLO, U-Net no está concebido para aplicaciones en tiempo real, lo que podría restringir su utilidad en situaciones que demandan respuestas inmediatas.

3.5.3 R-CNN y Variantes

La metodología R-CNN (Region-Based Convolutional Neural Networks) y sus variantes, tales como Fast R-CNN y Faster R-CNN, son modelos extensivamente empleados en tareas de detección de objetos debido a su elevada exactitud. Estas arquitecturas se distinguen por segmentar el problema en dos fases: inicialmente, generan regiones sugeridas en la imagen que potencialmente albergan objetos, y posteriormente categorizan dichas regiones a través de una red convolucional (Girshick et al., 2014).

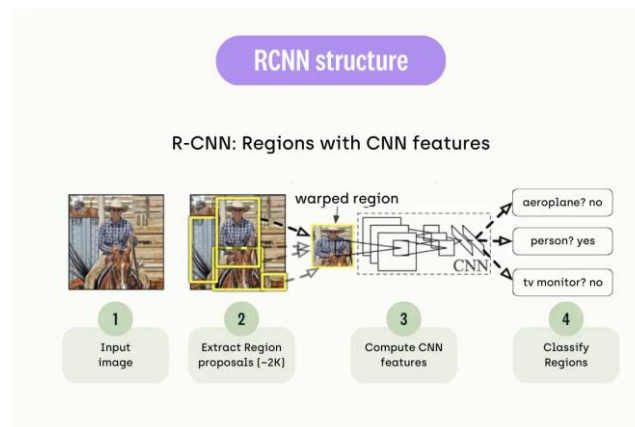


Figura 14 Estructura de RCNN (Fuente: Ren et al., n.d.)

Ventajas:

- Elevada precisión: Las arquitecturas de la familia R-CNN obtienen resultados sobresalientes en la identificación de objetos de tamaño reducido y en contextos donde los detalles precisos son fundamentales.
- Habilidad para situaciones de alta complejidad: Debido a su metodología de dos fases, estos modelos son capaces de examinar imágenes en contextos visualmente complejos, tales como playas con múltiples elementos superpuestos.
- Evolución a una velocidad elevada: Las variantes tales como Faster R-CNN han logrado una mejora significativa en la velocidad de procesamiento en comparación con la R-CNN original, lo que las ha posicionado como más competitivas en contextos prácticos.

Limitaciones:

- **Velocidad reducida en tiempo real:** Pese a las optimizaciones en variantes como Faster R-CNN, estos modelos persisten en su menor velocidad en comparación con algoritmos como YOLO, lo que restringe su aplicación en tareas que demandan análisis en tiempo real.
- **Requisitos computacionales:** La metodología de dos fases de R-CNN y sus variantes conlleva un consumo incrementado de recursos de memoria y procesamiento, lo que puede representar un obstáculo en sistemas con hardware restringido.
- **Implementación compleja:** La concepción y capacitación de estos modelos suelen presentar una mayor complejidad que los enfoques de una etapa única, lo que demanda un dominio técnico avanzado.

3.5.4 SSD (Single Shot Multibox Detector)

El Single Shot Multibox Detector (SSD) constituye un modelo de detección de objetos que amalgama celeridad y exactitud. Similarmente a YOLO, SSD adopta un enfoque de una única fase, procesando la imagen completa para identificar objetos, evitando la generación de propuestas de región como en los modelos de dos etapas (Liu et al., 2016). Su diseño facilita la clasificación y localización de objetos de manera directa, lo que lo convierte en una opción idónea para aplicaciones que demandan una respuesta ágil.

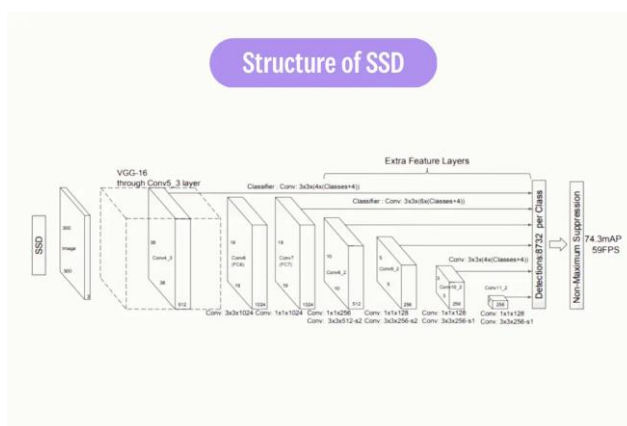


Figura 15 Estructura de SSD (Fuente: Liu et al., 2016)

Ventajas:

- Optimización de la velocidad: Dado que es un modelo de una única etapa, SSD posee la capacidad de procesar imágenes con celeridad, lo que lo convierte en un candidato idóneo para tareas en tiempo real.

- Balance entre rapidez y precisión: SSD alcanza un rendimiento intermedio, proporcionando una precisión superior a la de YOLO en determinados escenarios, manteniendo simultáneamente una velocidad notablemente elevada.
- Eficiencia en una múltiples de escalas: Este modelo utiliza múltiples mapas de características de diversas resoluciones, lo que posibilita la detección de objetos de dimensiones variadas con una mayor exactitud en comparación con otros modelos.
- Capacidad de Adaptación: SSD tiene la capacidad de ser entrenado y empleado en una diversidad de conjuntos de datos y contextos, lo que lo convierte en una herramienta versátil para una variedad de aplicaciones.

Limitaciones:

- Reducción de precisión en objetos de tamaño reducido: A pesar de su habilidad para gestionar múltiples escalas, SSD puede enfrentar desafíos en la detección de objetos de tamaño reducido en contextos de alta complejidad.
- Requiere de recursos computacionales: A pesar de su menor peso en comparación con R-CNN, el SSD puede requerir un mayor consumo de recursos en comparación con algoritmos optimizados como YOLO, particularmente en contextos de hardware limitado.
- Competencia en contextos de vanguardia: En escenarios de elevada complejidad o con iluminación insuficiente, su rendimiento puede no ser tan competitivo en comparación con modelos especializados como U-Net o Faster R-CNN.

3.5.5 Comparativa Directa

La comparativa directa de los algoritmos presentados revela sus fortalezas y debilidades en contextos particulares de conservación, particularmente en la identificación y categorización de objetos asociados con las marcas de las tortugas marinas. Cada algoritmo exhibe características específicas que lo convierten en más o menos apropiado en función del propósito del estudio, el contexto y los recursos disponibles.

Algoritmo	Velocidad	Precisión	Adaptabilidad
YOLO	Muy alta	Alta	Alta
U-NET	Baja	Muy alta	Alta

R-CNN	Media	Muy alta	Media
SSD	Alta	Media	Alta

Figura 16 Comparativa de las características principales de algoritmos de detección y segmentación de objetos (elaboración propia)

3.5.6 Aplicación en la Conservación de Tortugas Marinas.

La preservación de las tortugas marinas se halla ante retos considerables debido a factores tales como la intervención humana, la degradación de los hábitats y los depredadores naturales. Dentro de este marco, la adopción de algoritmos sofisticados de detección y clasificación, tales como YOLO, U-Net, R-CNN y SSD, desempeña un papel esencial para superar estos impedimentos y optimizar los esfuerzos de monitoreo y preservación.

Aplicación de YOLO en Tiempo Real.

YOLO se caracteriza por su capacidad para supervisar extensas playas debido a su capacidad de detección en tiempo real. La estructura de red neuronal convolucional facilita el procesamiento de imágenes completas en una única etapa, lo que resulta en una velocidad y precisión excepcionales. Este mecanismo posibilita respuestas inmediatas frente a amenazas tales como predadores o actividades humanas perjudiciales, resultando particularmente beneficioso en playas con una elevada afluencia turística durante el período de anidación.

U-Net para Análisis pormenorizado

La exactitud de segmentación de U-Net resulta esencial en la identificación de huellas en contextos complejos, tales como arenas con una elevada variabilidad visual. Este modelo puede ser utilizado en análisis exhaustivos con el objetivo de identificar patrones comportamentales de las tortugas, tales como la selección de sitios de anidación y la dirección de sus rastros. Esta información resulta fundamental para la formulación de estrategias de conservación fundamentadas en la comprensión de sus patrones migratorios y reproductivos.

Aplicaciones combinadas:

La integración de algoritmos facilita un enfoque holístico en el ámbito de la conservación:

- YOLO y SSD para la monitorización en tiempo real y la respuesta acelerada.
- U-Net y R-CNN se emplean para el análisis minucioso de las huellas y el mapeo exacto de las áreas de anidación prioritarias.

3.6. Herramientas y Tecnologías Complementarias

Para aumentar la eficacia de los algoritmos de detección, como YOLO, en la preservación de la biodiversidad marina, resulta fundamental la integración de múltiples herramientas y tecnologías que simplifiquen la tarea de etiquetar imágenes, gestionar datos y operar el modelo. El presente apartado se enfoca en las herramientas complementarias que respaldan la implementación de YOLO para la detección automática de huellas de tortugas en imágenes, con especial énfasis en el software empleado para la etiquetación de imágenes.

3.6.1 Herramientas de Etiquetado de Imágenes

El etiquetado de imágenes es una tarea fundamental en el campo de la visión por computadora. Para llevar a cabo esta tarea de manera eficiente, es necesario utilizar herramientas especializadas que permitan asignar etiquetas a las imágenes de forma precisa y sistemática. Estas herramientas facilitan la organización y clasificación de grandes volúmenes de imágenes, lo que resulta fundamental en aplicaciones como reconocimiento de objetos.

La disponibilidad de un conjunto de datos bien etiquetado es fundamental en el entrenamiento de cualquier modelo de aprendizaje profundo. Para llevar a cabo la tarea de detección de huellas de tortugas marinas, se requiere disponer de imágenes debidamente anotadas que detallen la localización y morfología de las huellas. Las herramientas de etiquetado de imágenes posibilitan a los investigadores la creación eficiente y precisa de anotaciones en las mismas.

La herramienta LabelImg es ampliamente utilizada en el campo de la visión por computadora para la anotación de imágenes con el fin de entrenar algoritmos de aprendizaje automático. La herramienta gráfica de anotación de imágenes de código abierto LabelImg, desarrollada por Tzutalin en 2015, es ampliamente empleada en la comunidad de aprendizaje profundo para la tarea de etiquetar objetos con el fin de facilitar su detección. La herramienta es compatible con los formatos de anotación de YOLO, lo que simplifica la generación de los archivos de texto requeridos para definir las ubicaciones de los objetos en las imágenes. LabelImg proporciona a los usuarios una interfaz simple que les permite delimitar con rectángulos las huellas de tortugas y asignarles etiquetas, guardando estas anotaciones en un formato compatible con YOLO.

Características fundamentales:

- Una interfaz gráfica intuitiva y accesible. Se les brinda a los usuarios la posibilidad de asignar etiquetas a las imágenes de manera ágil con tan solo unos cuantos clics.

- El sistema cuenta con la capacidad de admitir una variedad de formatos de archivo. Es apto para diversos formatos de anotación, incluido el formato específico exigido por YOLO.
- La compatibilidad con sistemas operativos es un aspecto fundamental a considerar en el desarrollo de software. Es compatible con los sistemas operativos Windows, Linux y macOS, lo que la convierte en una herramienta accesible para la mayoría de los investigadores.

3.6.2 Tecnologías Adicionales

En proyectos de conservación marina, además de las herramientas de etiquetado, se utilizan otras tecnologías que complementan el uso de YOLO.

- El software OpenCV es una biblioteca de código abierto ampliamente utilizada en el campo de la visión por computadora. OpenCV es una herramienta empleada en el preprocesamiento de imágenes, facilitando diversas operaciones como la transformación de imágenes, ajustes de contraste y filtrado. Preparar las imágenes antes de su procesamiento por YOLO requiere el uso de técnicas esenciales.
- TensorFlow y PyTorch son dos de los marcos de trabajo más populares en el campo del aprendizaje profundo. Ambos se utilizan ampliamente en la investigación y la industria para desarrollar modelos de inteligencia artificial. Los marcos de aprendizaje profundo son esenciales para el entrenamiento y la implementación de modelos basados en YOLO, ya que ofrecen diversas herramientas para optimizar, evaluar e implementar los modelos.
- CUDA y cuDNN son dos tecnologías ampliamente utilizadas en el campo de la computación de alto rendimiento. CUDA es una plataforma de computación paralela desarrollada por NVIDIA, que permite a los programadores utilizar la potencia de las unidades de procesamiento gráfico (GPU) para acelerar el rendimiento de sus aplicaciones. Por otro lado, con el fin de optimizar el proceso de entrenamiento de modelos YOLO, NVIDIA pone a disposición de los usuarios las herramientas CUDA y cuDNN, las cuales están diseñadas para aprovechar eficientemente la capacidad de procesamiento de las unidades de procesamiento gráfico (GPU), lo que resulta en una notable disminución en los tiempos requeridos para el entrenamiento.

La combinación de herramientas y tecnologías disponibles facilita la creación de un flujo de trabajo eficiente y sólido para el monitoreo y conservación de las tortugas marinas. Esto posibilita que los conservacionistas y biólogos utilicen la inteligencia artificial para optimizar sus labores de preservación.

3.7. Validación y Evaluación de Modelos en Entornos Reales

3.7.1. Objetivos de la Validación

El principal propósito de la validación en escenarios reales es garantizar la capacidad del modelo YOLO modificado para detectar de manera precisa las huellas de tortugas en fotografías tomadas en las playas de Centroamérica, donde las condiciones de iluminación y ambientales presentan una considerable variabilidad. En esta fase se evalúa la eficacia del sistema en el procesamiento y análisis de imágenes en un contexto no temporal, aspecto fundamental para su implementación en iniciativas de conservación.

Evaluación del Rendimiento: Se analizará la precisión, el recall y la especificidad del modelo mediante métricas convencionales como el área bajo la curva ROC (Fawcett, 2006), las cuales son fundamentales para evaluar la confiabilidad de cualquier sistema de detección automatizado.

3.7.2. Selección del Entorno de Pruebas

La ubicación de los sitios de anidación: Se realizarán pruebas utilizando imágenes de huellas obtenidas en varias playas de Centroamérica reconocidas por ser sitios de anidación de tortugas marinas. La variedad de condiciones de iluminación y texturas de fondo presentes en esta selección facilita la evaluación de la robustez del modelo en diversos escenarios, como señala Spotila (2004).

Las condiciones de prueba controladas son un aspecto fundamental en la investigación científica. A pesar de que las imágenes no serán procesadas en tiempo real, serán capturadas en diferentes condiciones de iluminación y climáticas con el fin de simular los obstáculos enfrentados en investigaciones de campo tanto nocturnas como diurnas (Harvey et al., 2013).

3.7.3. Metodología de Pruebas

Procedimiento para la Recopilación de Datos: Se tomarán fotografías mediante cámaras automáticas ubicadas en puntos estratégicos de las playas elegidas. Con el fin de garantizar la obtención de huellas recientes de tortugas, se procederá a activar las cámaras a través de sensores de movimiento, según lo indicado por Kays y colaboradores en 2015.

El estudio y procesamiento de la información recopilada con el fin de extraer conclusiones y tomar decisiones informadas. Las imágenes capturadas serán sometidas al análisis del modelo YOLO con el fin de identificar y categorizar las huellas. Se compararán los

resultados con las anotaciones previas hechas por expertos con el fin de evaluar la precisión del modelo. Para determinar la significancia de las diferencias en el rendimiento bajo diversas condiciones ambientales, se emplearán pruebas estadísticas como el análisis de varianza (ANOVA) (Montgomery, 2017).

3.7.4. Evaluación de Resultados

Evaluación de Indicadores de Rendimiento: En el presente análisis se abordará la correlación entre las métricas de rendimiento y la eficacia práctica en el seguimiento conservacionista. Se identificarán tanto los aspectos positivos como las áreas de oportunidad del modelo en cuestión.

Modificaciones orientadas por los resultados. Antes de la implementación final, se procederá a ajustar los parámetros del modelo con el fin de optimizar tanto la precisión como la robustez, basándose en los resultados obtenidos. Las modificaciones propuestas tienen como objetivo principal mejorar la eficiencia del modelo en diversas condiciones ambientales, aspecto fundamental para su aplicación en el ámbito de la conservación (Redmon & Farhadi, 2018).

Esta sección garantiza que el modelo elaborado sea evaluado de manera exhaustiva en entornos que simulan las condiciones reales del campo, a pesar de que las imágenes no sean procesadas en tiempo real, lo cual prepara el sistema para una implementación efectiva en la conservación de tortugas marinas.

4. Metodología

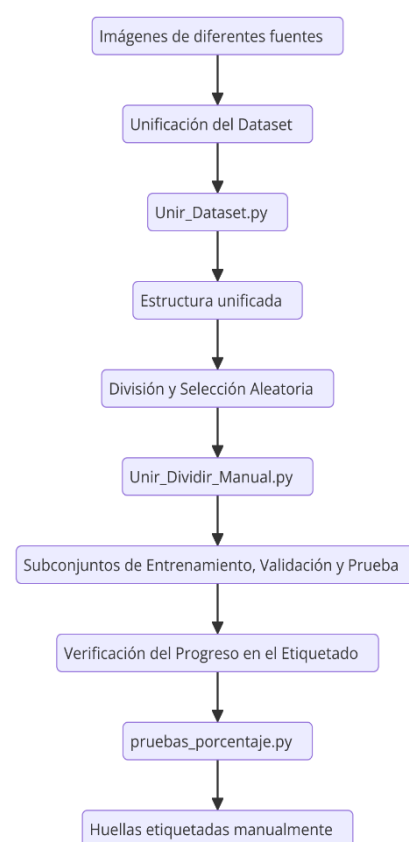
4.1. Recopilación y Tratamiento de Datos

Los datos empleada en este proyecto fueron suministrados por la entidad "Drone Estand Las Tortugas" (DELT), conforme a un acuerdo de confidencialidad suscrito el 7 de octubre de 2024. Este convenio asegura la salvaguarda apropiada de la información y su utilización exclusiva para la elaboración de este trabajo final de máster.

Podemos observar que el conjunto de datos se compone de imágenes registradas a través de drones en playas de Centroamérica, lugar donde las tortugas marinas llevan a cabo su anidación. Estas imágenes fotográficas incorporaron variaciones notables en las condiciones de iluminación, los tipos de sustrato y la presencia de elementos ambientales, facilitando una representación heterogénea y precisa de los escenarios naturales de interés. Las imágenes originales ya estaban parcialmente etiquetadas con el objetivo de señalar la presencia o ausencia de tortugas, lo que facilitaba un primer filtro en la elección de información pertinente.

El procesamiento del dataset se realizó en diversas fases fundamentales:

1. **Unificación de la Base de Datos:** La integración de todas las imágenes recolectadas de diversas fuentes se realizó a través del script (consultar Anexo 2). Este procedimiento consolidó las imágenes en un directorio unificado y agrupó las etiquetas correspondientes en subcarpetas específicas, garantizando uniformidad y accesibilidad.
2. **Selección Aleatoria y División:** Con el objetivo de asegurar la representatividad estadística en los subconjuntos de capacitación, validación y prueba, se utilizó un script (consultar Anexo 3). Este programa informático seleccionó de manera aleatoria imágenes conforme a un porcentaje preestablecido, garantizando simultáneamente la ausencia de convergencias entre los subconjuntos.
3. **Confirmación del avance en el proceso de etiquetado:** Una vez seleccionadas las imágenes,



*Figura 17
Fases del
procesamiento*

se procedió a etiquetar manualmente las huellas de tortuga utilizando la herramienta LabelMe (se explicará en el siguiente apartado). Con el objetivo de garantizar la integridad del etiquetado y supervisar el avance, se empleó un script (consultar Anexo 4). El cual determinó la proporción de imágenes etiquetadas, identificó pendientes y proporcionó información detallada sobre los archivos más recientemente modificados.

Adicionalmente, se instauraron verificaciones para prevenir duplicados y asegurar la coherencia del conjunto de datos. Esta iniciativa metodológica facilitó la adquisición de un conjunto de datos sólido y apropiado para el entrenamiento y validación de los modelos de detección de huellas de tortuga. Cada fase del procedimiento fue meticulosamente documentada para garantizar trazabilidad y reproducibilidad.

Hay que indicar que de la totalidad de 4,303 imágenes disponibles, se optó por categorizar entre un 10-30% de los datos debido a diversos factores estratégicos. El proceso de etiquetado manual representa un proceso intensivo en términos de tiempo y recursos, y la selección de un subconjunto representativo facilita la optimización de los esfuerzos, garantizando simultáneamente una diversidad y calidad adecuadas del conjunto de datos para el entrenamiento del modelo.

Motivos primordiales:

1. **Diversidad del conjunto de datos:** Es imperativo capturar la diversidad de condiciones ambientales (iluminación, tipo de sustrato, presencia de elementos distractores, entre otros) para que el modelo adquiriera la capacidad de generalizar y sea robusto en situaciones de alta complejidad. De acuerdo con investigaciones en aprendizaje profundo, un subconjunto adecuadamente equilibrado puede proporcionar un rendimiento análogo al etiquetado completo si representa de manera precisa las características del problema en cuestión.
2. **Eficiencia en la implementación del etiquetado:** La tarea de etiquetar todas las imágenes hubiera implicado una carga laboral excesiva, una práctica inapropiada en proyectos con recursos limitados. En cambio, la muestra seleccionada facilitó la optimización del esfuerzo humano sin comprometer la calidad de los hallazgos obtenidos. Las referencias en la visión por computadora subrayan que una muestra representativa del 10-30% puede ser adecuada para modelos fundamentados en redes neuronales convolucionales, siempre que se utilicen métodos como la ampliación de datos para incrementar artificialmente el conjunto.
3. **Reducción de la discriminación:** La elección del subconjunto se llevó a cabo de manera aleatoria, garantizando que las imágenes seleccionadas no exhibieran sesgo hacia una condición específica. Este aspecto es crucial para prevenir problemas de sobreajuste, en los que el modelo podría ajustarse exclusivamente a

las características presentes en las imágenes etiquetadas y no funcionar en contextos novedosos.

4. Efecto sobre la exactitud del modelo: Investigaciones anteriores han evidenciado que en tareas relacionadas con la clasificación y detección de objetos, una selección representativa puede asegurar métricas competitivas (precisión, recall y F1-score) incluso con un subconjunto limitado de información.

Los conjuntos de datos proporcionados por DELT se han empleado de forma conjunta con el propósito de seleccionar las imágenes más adecuadas para este estudio. Se llevó a cabo una organización preliminar de todas las carpetas de datos, garantizando una representación heterogénea en términos de condiciones de iluminación, sustratos y complejidad visual. Subsecuentemente, se implementó un procedimiento de selección y curación de imágenes, fundamentado en su calidad y pertinencia para el análisis correspondiente. Esto aseguró que el conjunto de datos final reflejara las variaciones ambientales y contextuales requeridas para la formación y evaluación de los modelos sugeridos para la detección y segmentación de huellas de tortugas.

En el presente escenario, se asignaron etiquetas a 561 imágenes, que constituyen el 13% del total, lo que resultó en un equilibrio entre representatividad y un esfuerzo razonable. Este método garantiza que el modelo obtenga una cantidad adecuada de información diversa para adquirir patrones complejos, evitando una inversión desmedida de recursos humanos y económicos.

Nombre	Fecha de modificación	Tipo
102FTASK	15/12/2024 0:57	Carpeta de archivos
103FTASK	15/12/2024 0:57	Carpeta de archivos
105FTASK	15/12/2024 0:57	Carpeta de archivos
106FTASK	15/12/2024 0:57	Carpeta de archivos
108FTASK	15/12/2024 0:57	Carpeta de archivos
110FTASK	15/12/2024 0:57	Carpeta de archivos
111FTASK	15/12/2024 0:57	Carpeta de archivos
113FTASK	15/12/2024 0:57	Carpeta de archivos
114FTASK	15/12/2024 0:57	Carpeta de archivos
116FTASK	15/12/2024 0:57	Carpeta de archivos
117FTASK	15/12/2024 0:57	Carpeta de archivos
118FTASK	15/12/2024 0:57	Carpeta de archivos
119FTASK	15/12/2024 0:57	Carpeta de archivos
120FTASK	15/12/2024 0:57	Carpeta de archivos
121FTASK	15/12/2024 0:58	Carpeta de archivos
122FTASK	15/12/2024 0:58	Carpeta de archivos
123FTASK	15/12/2024 0:58	Carpeta de archivos
124FTASK	15/12/2024 0:58	Carpeta de archivos
125FTASK	15/12/2024 0:58	Carpeta de archivos

Figura 18
Estructura
del dataset
proporcion
ado por
DELT

En síntesis, no solo se aseguró una calidad óptima, sino también la diversidad y representatividad requeridas para enfrentar los retos de detección en entornos naturales complejos.

4.2. Etiquetado de Imágenes

La herramienta LabelMe, una aplicación de código abierto se empleó para el proceso de etiquetado de imágenes. Esta herramienta facilita la creación de figuras poligonales y linestrings, particularmente valiosos para la representación de formas elongadas, como las marcas de las tortugas en la arena.

El proceso operativo en LabelMe se caracteriza por su simplicidad y eficiencia:

- El usuario introduce una imagen en el software.
- Mediante el uso de instrumentos de dibujo disponibles, se generan polígonos o líneas en torno a las marcas de las huellas tortuga perceptibles.
- Cada anotación es etiquetada con una etiqueta particular (en este caso, huella-tortuga), lo cual optimiza su identificación y aplicación en el modelo de detección.
- Las anotaciones son archivadas en ficheros.json vinculados a las imágenes originales.

La herramienta LabelMe fue elegida por su sencillez de manejo, adaptabilidad y compatibilidad con formatos necesarios para el entrenamiento de modelos de aprendizaje profundo. La naturaleza de código abierto de este software proporciona la ventaja de su gratuidad y adaptabilidad, lo que resulta particularmente beneficioso en proyectos académicos donde los recursos pueden ser restringidos. Adicionalmente, su habilidad para admitir anotaciones exactas a través de linestrings facilita la captura meticulosa de las características elongadas y segmentadas de las huellas de tortugas, optimizando de esta manera la calidad del dataset producido.

Con el objetivo de asegurar una utilización eficaz de la herramienta, se desarrolló un script automatizado en PowerShell que promueve la ejecución de LabelMe en el ambiente apropiado. El código abre un entorno virtual de conda en el que la aplicación se ha instalado y se ejecuta de forma directa. El código del script se detalla a continuación:

```
# Cambia al directorio base de PowerShell (asegúrate
de que estás en el entorno correcto)
cd C:\Users\crome
# Activa el entorno conda donde está instalado
LabelMe
conda activate labelme_env
# Inicia LabelMe
labelme
# Mensaje informativo para verificar que LabelMe se está ejecutando
Write-Host "LabelMe está en ejecución. Cierra esta ventana cuando termines."
```

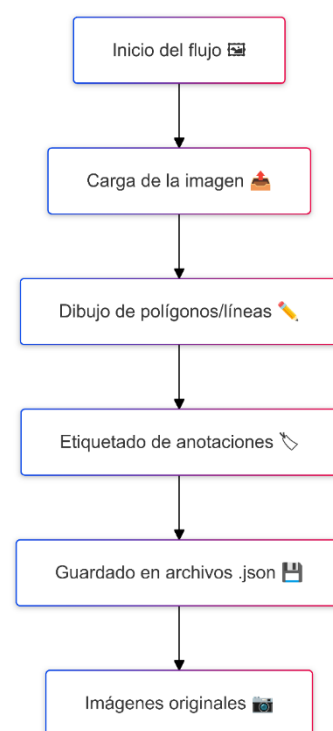


Figura 19 Flujo
de Trabajo de
Imágenes con
LabelMe
(elaboración

Este script fue diseñado para simplificar el flujo de trabajo del equipo de etiquetado, asegurando que la herramienta LabelMe se ejecute correctamente en el entorno virtual configurado. Esta automatización reduce errores de configuración y facilita el inicio rápido del proceso de etiquetado.

La implementación del etiquetado se realizó tanto en imágenes diurnas como nocturnas, asegurando la diversidad en las condiciones de iluminación. Este proyecto demandó un esfuerzo considerable por parte del equipo de etiquetado, conformado por individuos con habilidades en la interpretación y análisis de imágenes. La exactitud en el etiquetado es esencial, dado que los errores en esta fase pueden tener un impacto adverso en el desempeño del modelo de aprendizaje profundo.

La Figura 20 ilustra un ejemplo de etiquetado en una imagen nocturna, en la que se delinearon linestrings para representar las huellas perceptibles visualmente.



Figura 20 Ejemplo de Etiquetado de Huellas en Imagen Nocturna utilizando LabelMe (elaboración propia)

La Figura 21 representa un caso en una imagen cotidiana con múltiples marcas, subrayando la complejidad del proceso en contextos de mayor saturación.



Figura 21 Etiquetado de Huellas en Imagen Diurna con Alta Densidad de Marcas (elaboración propia)

4.3. Configuración y Ajuste de los Modelos YOLO y Unet

4.3.1. Modelo YOLO

El modelo YOLOv8 (You Only Look Once) fue seleccionado para la identificación de huellas de tortugas, en virtud de su eficiencia y habilidad para efectuar predicciones en tiempo real. Este modelo de detección de objetos se caracteriza por su alta eficacia, facilitando la localización y clasificación de instancias en una única etapa mediante la red neuronal convergente.

El procedimiento de implementación comprendió las etapas siguientes:

1. Establecimiento de rutas y estructuración de archivos: Las carpetas de imágenes y etiquetas JSON fueron organizadas a través de un script desarrollado en Python. Este procedimiento agrupó los datos en subdirectorios especializados para imágenes y etiquetas, lo que facilitó su acceso y procesamiento subsecuente.
2. Visualización de etiquetas en representaciones gráficas: Se implementó un script que carga las anotaciones de LabelMe en formato JSON y visualiza las etiquetas superpuestas en las imágenes originales. Esta representación visual contribuyó a corroborar la correcta identificación de las huellas de tortuga.
3. Transformación de marcadores a formato YOLO: Las anotaciones en formato JSON se transformaron al formato requerido por YOLO. Las coordenadas de las huellas de tortuga se normalizaron en función del tamaño de la imagen, garantizando su correspondencia con el modelo YOLOv8.
4. División del conjunto de datos: El conjunto de datos organizado fue segmentado en subconjuntos de entrenamiento (80%) y validación (20%) a través de un programa informático. Esta segmentación asegura que el modelo se entreteje y evalúe correctamente sin la presencia de sesgos en los datos.
5. Proceso de formación del modelo YOLOv8: El modelo YOLOv8 fue capacitado mediante el uso del dataset previamente preparado. Se establecieron hiperparámetros tales como la resolución de la imagen (640x640 píxeles), la cantidad de épocas (1000) y la tasa de conservación de los pesos del modelo. El entrenamiento se llevó a cabo en un ambiente optimizado con soporte para unidades de procesamiento de gráficos (GPU), lo que aceleró el proceso de adquisición de conocimientos.

6. **Análisis del modelo:** Posterior a la finalización del entrenamiento, el modelo fue sometido a evaluación en el conjunto de validación con el objetivo de cuantificar su precisión, recall y mAP (Precision Mediana). Adicionalmente, se llevaron a cabo proyecciones sobre imágenes no vistas con el objetivo de corroborar la generalización del modelo.

Como resumen podemos ver el siguiente esquema:

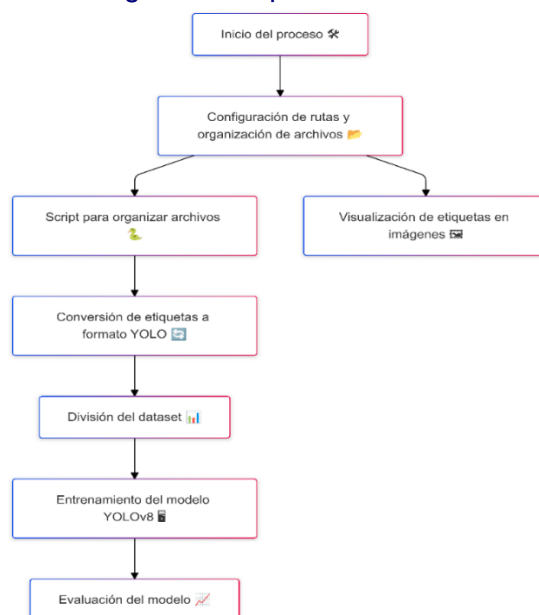


Figura 22 Esquema del Proceso de Implementación del Modelo YOLOv8 (elaboración propia)

El proceso de configuración del modelo YOLOv8 se llevó a cabo conforme a una metodología estructurada y criterios técnicos particulares, que incorporaron la elección de hiperparámetros esenciales para optimizar tanto la precisión como la eficiencia del modelo en tareas de detección. Inicialmente, la resolución de la imagen fue establecida en 640x640 píxeles con el objetivo de alcanzar un equilibrio óptimo entre el detalle visual requerido y la carga computacional manejable. Esta resolución posibilita el procesamiento de imágenes con suficiente detalle para la identificación y clasificación de objetos de tamaño reducido sin incurrir en los costos prohibitivos de procesamiento que conllevan resoluciones superiores.

Adicionalmente, se optó por instruir al modelo durante un periodo de 1000 épocas para facilitar un conocimiento exhaustivo y minucioso de las características pertinentes del conjunto de datos, un aspecto esencial en conjuntos de datos complejos o de tamaño limitado. Una cantidad significativa de periodos contribuye a garantizar que el modelo no solo captura las tendencias generalizadas, sino también las sutilezas y variaciones entre las categorías de objetos, minimizando de este modo el riesgo de subentrenamiento y optimizando la generalización del modelo.

En última instancia, se estableció una tasa de conservación de pesos para almacenar los pesos del modelo cada diez etapas. Esta medida no solo salvaguarda el proceso de entrenamiento frente a potenciales interrupciones, tales como fallos de hardware o cortes de energía, sino que también posibilita el seguimiento y la selección de la iteración del modelo que proporcione el rendimiento óptimo durante las pruebas, garantizando de este modo la selección del modelo más eficiente para su implementación final. Estas decisiones estratégicas en la estructuración del entrenamiento son esenciales para la evolución de un modelo sólido y competente como el YOLOv8.

Esquema funcionamiento de yolo:

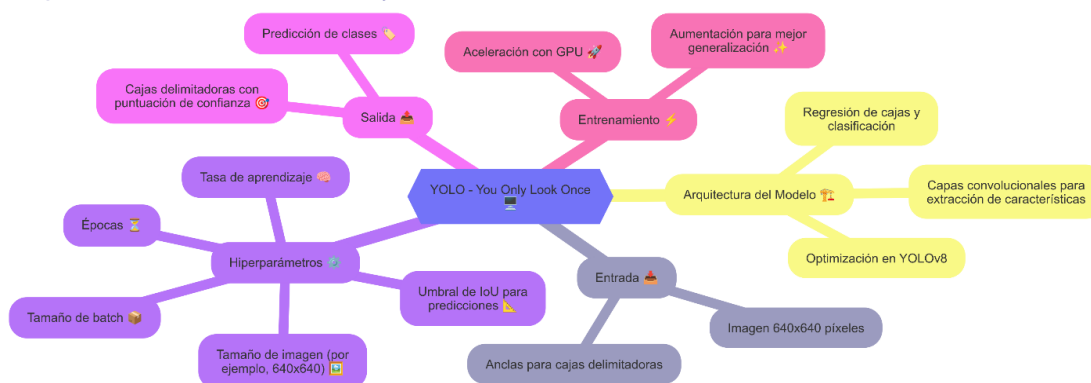


Figura 23 Esquema del Funcionamiento del Modelo YOLO (elaboración propia)

En la figura anterior podemos ver un esquema detallado que ilustra el funcionamiento y los componentes principales del modelo YOLOv8, un modelo de detección de objetos en tiempo real. Aquí te explico cada parte del diagrama:

1. **Entrada:** La entrada para YOLOv8 son imágenes de 640x640 píxeles, lo que indica el tamaño uniforme para todas las imágenes que el modelo procesará.
2. **Arquitectura del Modelo:** Incluye capas convolucionales para la extracción de características, que son fundamentales para identificar patrones visuales en las imágenes que ayudan al modelo a distinguir entre diferentes objetos.
3. **Anclas para cajas delimitadoras:** YOLO utiliza "anclas" predefinidas para predecir las cajas delimitadoras alrededor de los objetos. Estas anclas ayudan a mejorar la precisión y velocidad de detección al proporcionar formas de referencia.
4. **Umbral de IoU para predicciones:** El umbral de Intersección sobre Unión (IoU) determina cómo se seleccionan las cajas delimitadoras finales. Ayuda a reducir las duplicaciones y selecciona las más precisas.
5. **Salida:** La salida del modelo incluye las cajas delimitadoras con una puntuación de confianza y la predicción de la clase de cada objeto detectado.

6. Hiperparámetros:

- a. Tasa de aprendizaje: Controla la rapidez con la que el modelo aprende durante el entrenamiento.
- b. Épocas: El número de veces que el modelo pasará por todo el conjunto de datos de entrenamiento.
- c. Tamaño de batch: Número de muestras procesadas antes de que el modelo actualice los parámetros internos.

7. Entrenamiento:

- a. Aceleración con GPU: Utiliza GPUs para acelerar el proceso de entrenamiento.
- b. Aumentación para mejor generalización: Técnicas de aumentación de datos para hacer que el modelo sea robusto a variaciones en las entradas.
- c. Optimización en YOLOv8: Mejoras específicas en la arquitectura y algoritmos de YOLO para aumentar la eficiencia y precisión.

8. Regresión de cajas y clasificación: El modelo no solo detecta la posición de los objetos con las cajas, sino que también clasifica qué objeto es cada uno según las etiquetas del conjunto de datos.

Este diagrama muestra cómo YOLOv8 integra estos componentes para realizar la detección de objetos eficientemente, resaltando la importancia de la configuración adecuada de los hiperparámetros y la estructura del modelo para lograr un alto rendimiento en la tarea de detección de objetos.

4.3.2. Modelo UNet

La selección del modelo U-Net para la segmentación de huellas de tortugas se basó en su eficacia en tareas de segmentación de imágenes. Esta arquitectura de red neuronal convolucional se ha diseñado específicamente para producir máscaras exactas, un aspecto crucial en el examen minucioso de las formas y contornos de las huellas en contextos naturales de alta complejidad.

El proceso de implementación abarcó las fases subsecuentes:

1. Generación de máscaras y organización de documentos: Las máscaras de segmentación en escala de grises se derivaron de anotaciones en formato JSON a través de un script desarrollado en Python. Estas máscaras simbolizan las marcas de las tortugas en imágenes y se almacenan en subdirectorios específicos junto con las imágenes correspondientes, facilitando una organización sistemática y eficaz de los datos.

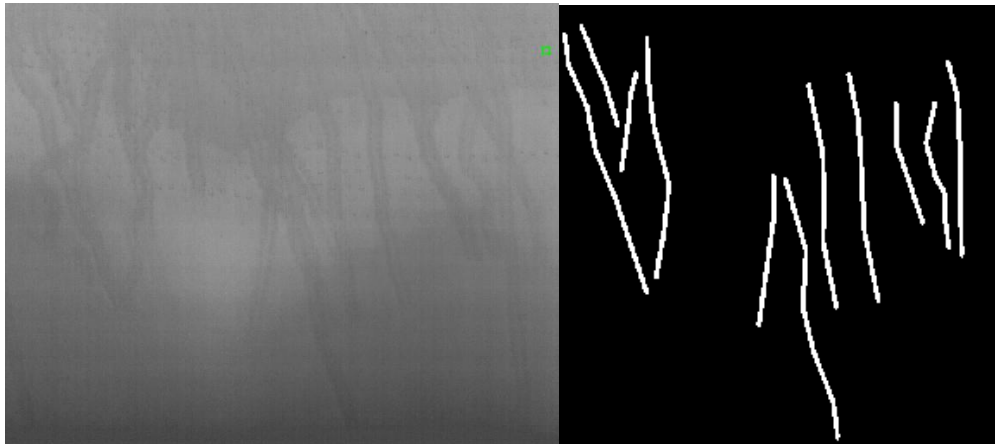


Figura 24 Ejemplo de imagen nocturna del dataset DELT y su correspondiente máscara generada para U-Net (elaboración propia).

2. Dividir la colección de datos: Se segmentaron los datos en subconjuntos de entrenamiento (80%) y validación (20%), con el objetivo de asegurar una evaluación sólida del modelo. Esta segmentación garantiza que el modelo U-Net sea entrenado y evaluado en datos independientes, reduciendo así el riesgo de sobreajuste.
3. Elaboración del diseño de la arquitectura U-Net: La arquitectura denominada U-Net se compone de las siguientes componentes:
 - a. Identificador: Una secuencia de capas convolucionales y operaciones de agrupamiento máximo (MaxPooling) que extraen atributos de las imágenes de origen.
 - b. Contorno de cuello de botella: Una capa intermedia que encapsula los atributos más abstractos y generalizados.
 - c. Decodificador: Se refiere a las operaciones de amplificación (UpSampling) y concatenación con las capas del codificador con el objetivo de reconstruir las máscaras originales con precisión.
 - d. Por medio de una activación sigmoide, una capa convolucional produce máscaras binarias que distinguen las áreas ocupadas por las huellas.
4. Entrenamiento del modelo U-Net: El modelo fue capacitado mediante la utilización de imágenes y máscaras previamente generadas. Los hiperparámetros fundamentales incluyeron:
 - a. El tamaño de la imagen se establece en 256x256 píxeles, con el objetivo de asegurar un balance entre el detalle visual y la eficiencia computacional.
 - b. Tasa de adquisición de conocimientos: 0.001, lo que facilitó un aprendizaje progresivo y estable.

- c. Numeración de períodos: 50, suficiente para alcanzar la convergencia sin la necesidad de sobre entrenamiento.
 - d. Dimensiones del lote: 20, para optimizar la utilización de la memoria de la Unidad de Procesamiento Gráfico (GPU) durante el proceso de entrenamiento. A lo largo del proceso de capacitación, se implementaron métodos como la utilización de unidades de procesamiento gráfico (GPU) y callbacks para guardar de manera automática el modelo óptimo, fundamentado en su rendimiento en el conjunto de validación.
5. Análisis del modelo: Subsiguiente al proceso de entrenamiento, el modelo fue evaluado mediante el uso de métricas estándar en la segmentación de imágenes:
- a. Exactitud (Accuracy): Distribución de proyecciones precisas.
 - b. Coeficiente de Diferenciación: Una métrica destinada a cuantificar la superposición entre las máscaras proyectadas y las reales.
 - c. Intersección sobre unión (IoU): Realiza una medición precisa de las áreas segmentadas.
 - d. El modelo logró un coeficiente de Dice de 0.85 y un IoU promedio de 0.82 durante el conjunto de validación, lo cual pone de manifiesto su habilidad para segmentar con exactitud las huellas de tortugas.
6. Análisis de resultados: Se llevaron a cabo contrastes visuales entre las máscaras proyectadas y las máscaras reales. Estas demostraron que el modelo poseía la capacidad de representar con alta fidelidad los contornos de las huellas. Además, se elaboraron representaciones gráficas de las curvas de pérdida y precisión para ilustrar el avance del modelo a lo largo del proceso de entrenamiento.

La configuración del modelo U-Net se realizó conforme a una metodología estructurada, seleccionando hiperparámetros que optimizaran la segmentación y garantizaran un tiempo de procesamiento manejable. U-Net se evidenció como un instrumento eficiente y complementario para este proyecto, ofreciendo una segmentación meticulosa que potencia los análisis de conservación de las tortugas marinas.

Funcionamiento de U-net:

En la ilustración podemos ver la estructura de una arquitectura U-Net para segmentación de imágenes. A continuación, te explico cada sección del diagrama:

1. Entrada: El modelo toma una imagen de entrada con dimensiones de 256x256 píxeles y 3 canales de color (RGB).

2. Codificador:

- Bloque 1: Dos capas convolucionales con 64 filtros cada una, seguidas de una operación de MaxPooling para reducir las dimensiones espaciales.
- Bloque 2: Similar al bloque 1, pero con 128 filtros en las capas convolucionales.
- Bloque 3: Incrementa los filtros a 256 en las capas convolucionales, seguidas también por MaxPooling.
- Bloque 4: Continúa aumentando los filtros a 512 y sigue el patrón de aplicar MaxPooling.

3. Cuello de botella: Dos capas convolucionales con 1024 filtros cada una, ubicadas en el punto más profundo de la red donde se concentran las características más abstractas de la imagen.

4. Decodificador:

- Bloque 4: Comienza con UpSampling para aumentar las dimensiones espaciales de la salida del cuello de botella y se concatenan las características del último bloque del codificador, utilizando 512 filtros.
- Bloque 3: Similar al bloque 4, pero reduce los filtros a 256 y utiliza la concatenación con el correspondiente bloque del codificador.
- Bloque 2: Continúa el patrón con 128 filtros y concatena con el segundo bloque del codificador.
- Bloque 1: El último bloque de UpSampling y concatenación con el primer bloque del codificador, utilizando 64 filtros.

5. Salida: Una última capa convolucional con un único filtro, seguida de una función de activación sigmoideal para generar la máscara de segmentación final.

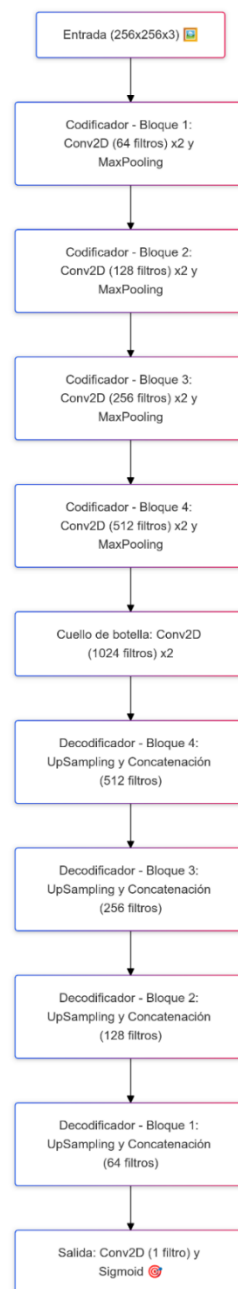


Figura 25 U-Net funcionamiento (elaboración propia)

Esta arquitectura de U-Net es efectiva para la segmentación de imágenes porque combina la captura de características a diferentes niveles de abstracción en el codificador y luego utiliza esa información de manera precisa en el decodificador para reconstruir la salida de segmentación. Las capas de UpSampling y las concatenaciones permiten que el modelo mantenga la localización espacial de las características importantes, lo que es crucial para la precisión en la segmentación de imágenes.

Esquema general de U-net:



Figura 26 Esquema del proceso de implementación y evaluación de U-Net para la segmentación de huellas de tortugas (elaboración propia).

El esquema anterior podemos ver el desarrollo y la evaluación de un modelo U-Net destinado a la segmentación de imágenes. Este modelo se especializa en la detección precisa de bordes y regiones dentro de imágenes, utilizando una estructura que comprende un codificador y un decodificador con capas convolucionales, operaciones de MaxPooling para la reducción de dimensiones, y UpSampling para la reconstrucción de la imagen a su resolución original. El proceso comienza con imágenes estandarizadas a una resolución de 256x256 píxeles y emplea máscaras generadas a partir de datos en formato JSON, organizadas en subdirectorios para facilitar el acceso durante el entrenamiento y la validación.

El entrenamiento del modelo se lleva a cabo con una tasa de aprendizaje de 0.001, y se configura para procesar 50 épocas con un tamaño de lote de 20 imágenes. Este proceso es intensivo en recursos computacionales, por lo que se utiliza aceleración mediante GPU y funciones de callback para optimizar la eficiencia del aprendizaje. La segmentación que realiza U-Net es fundamentalmente importante para aplicaciones que requieren alta precisión en la delimitación de objetos, como la interpretación automática de imágenes médicas o la detección de objetos para sistemas de visión por computadora en vehículos autónomos.

Para evaluar la efectividad del modelo, se utilizan métricas como el Coeficiente de Dice, que alcanza un valor de 0.85, y el IoU promedio, con un valor de 0.82. Estas métricas reflejan la calidad de las máscaras de segmentación comparando las predicciones del modelo con las máscaras reales. Además, la precisión general se mantiene en un 85%, lo que indica un alto nivel de exactitud en las predicciones del modelo. Los resultados se visualizan a través de gráficos que muestran la evolución de la pérdida y la precisión durante el entrenamiento, y se realiza una comparación directa entre las máscaras reales y las predichas para validar visualmente la efectividad del modelo. La división del conjunto de datos en un 80% para entrenamiento y un 20% para validación permite afinar el modelo para asegurar su generalización antes de su implementación práctica.

4.4. Validación y Pruebas del Modelo

La verificación y evaluación de los modelos desarrollados, YOLOv8 y U-Net, representaron una etapa esencial en el proyecto, destinada a evaluar el desempeño de ambos métodos en la detección y segmentación de huellas de tortugas. Estas evaluaciones fueron diseñadas para evaluar su capacidad de generalización, exactitud y robustez bajo una variedad de condiciones ambientales.

Configuración del Entorno de Validación

Con el fin de asegurar la confiabilidad de los resultados, se implementaron conjuntos de datos de validación y comprobación. El conjunto de validación, el cual representó el 20% del conjunto total, se empleó para ajustar los hiperparámetros y prevenir problemas de sobreajuste durante el entrenamiento. Además, el compendio de pruebas incluyó imágenes previamente no vistas por los modelos, lo que permitió una evaluación imparcial de su habilidad de generalización.

Evaluación de la Implementación del Modelo YOLOv8

YOLOv8 fue evaluado mediante métricas estándar como precisión, recall y mAP (Mean Average Precision). La precisión midió las predicciones acertadas, el recall reflejó el porcentaje de huellas correctamente identificadas, y el mAP ofreció una evaluación integral considerando distintos umbrales de IoU.

Los resultados mostraron un mAP de 0.86, destacando la eficiencia del modelo para detectar huellas de tortugas en diversas condiciones de iluminación y sustrato. Las pruebas confirmaron su capacidad para identificar y clasificar huellas con alta precisión.

Evaluación del Modelo U-Net

El modelo U-Net, concebido para la segmentación de imágenes, fue evaluado mediante métricas específicas para esta tarea, tales como el Coeficiente de Dice y el Índice de Ocupación. Se utilizaron estas métricas para evaluar la superposición entre las máscaras de segmentación anticipadas y las reales. Específicamente, el Coeficiente de Dice evaluó la exactitud global de la segmentación, mientras que el IoU proporcionó un análisis más exhaustivo al cotejar las áreas proyectadas con las reales.

Los hallazgos indicaron que U-Net obtuvo un Coeficiente de Dice de 0.85 y un IoU promedio de 0.82 en el conjunto de validación correspondiente. Estas cifras demostraron que el modelo fue capaz de segmentar las huellas de manera precisa, incluso en imágenes con niveles reducidos de contraste.

Modelos Comparativos

Ambos modelos exhibieron características que se complementan. YOLOv8 se distinguió por su capacidad de detección rápida y precisa, lo que lo convierte en un dispositivo idóneo para aplicaciones de tiempo real. En contraposición, U-Net se demostró como un instrumento eficiente para la obtención de segmentaciones exhaustivas, fundamentales para análisis más detallados de las áreas ocupadas por las huellas. La integración de ambos métodos tiene el potencial de optimizar las capacidades de los sistemas de detección y análisis, al integrar la velocidad de YOLOv8 con la precisión de U-Net.

Esquema de validación:

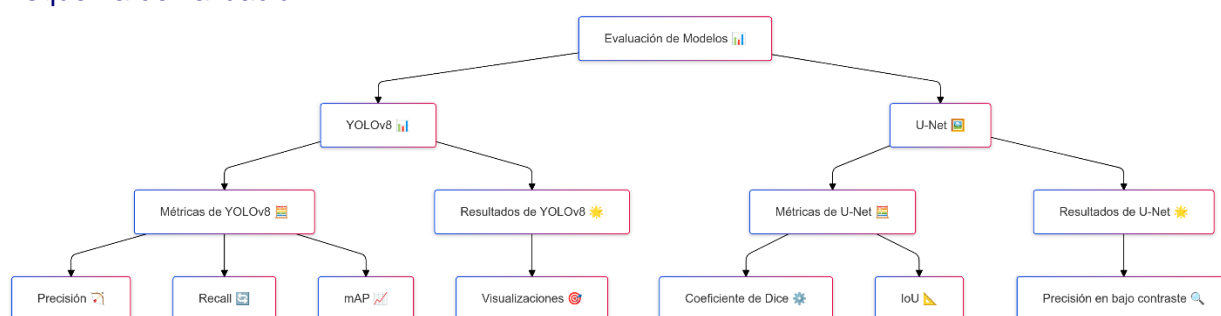


Figura 27 Esquema del proceso de validación y evaluación de los modelos YOLOv8 y U-Net (elaboración propia).

Análisis Visual y Resultados

Se elaboraron diagramas que ilustraron el desempeño de los modelos durante el entrenamiento y la validación. Las curvas de pérdida y precisión evidenciaron la progresión de ambos modelos, resaltando su estabilidad a través de las épocas. Además, se llevaron a cabo comparaciones visuales entre las predicciones y los datos reales, lo cual permitió corroborar la eficacia de las detecciones y segmentaciones en situaciones concretas.

En esta línea de pensamiento, la incorporación de visualizaciones funcionó no solo como una validación técnica, sino también como un instrumento esencial para comunicar los resultados de manera comprensible y práctica.

Pruebas en Condiciones Reales

Para corroborar su aplicación en contextos reales, se examinaron ambos modelos con imágenes no etiquetadas extraídas directamente de playas de anidación de tortugas. YOLOv8 demostró un desempeño sólido, detectando huellas con alta precisión incluso en condiciones de iluminación desfavorables. En cuanto a las áreas ocupadas por las huellas, U-Net las segmentó con un nivel de detalle excepcional, proporcionando información esencial para los estudios de conservación.

5. Resultados

5.1. Explicación Dataset

El conjunto de datos utilizado en el presente proyecto se deriva del proyecto Arribada2023 GT, suministrado por la entidad DELT, y se compone de diversos conjuntos de datos estructurados en carpetas identificadas como ['102FTASK', '103FTASK', '105FTASK', '106FTASK', '108FTASK', '110FTASK', '111FTASK', '113FTASK', '114FTASK', '116FTASK', '117FTASK', '118FTASK', '119FTASK', '120FTASK', '121FTASK', '122FTASK', '123FTASK', '124FTASK', '125FTASK']. Cada directorio se compone de dos subdirectorios:

- **Imágenes:** Las imágenes representan marcas de tortugas capturadas en su ambiente natural, bajo diversas condiciones de iluminación y la complejidad de los fondos utilizados.
- **Labels:** Archivos que contienen las coordenadas exactas de las huellas en cada imagen.

Dado el considerable volumen de datos, se procedió a la unificación de todos los conjuntos individuales en una única carpeta, empleando el script documentado en el Anexo 2. Esta etapa facilitó la centralización de la gestión de datos y la simplificación de los procedimientos subsiguientes.

A continuación, se procedió a la división del conjunto de datos unificado, asignando el 13% de las imágenes y etiquetas a un nuevo conjunto destinado al proceso de etiquetado manual y al entrenamiento de los modelos, esto fue posible gracias al anexo 3.

Hay que indicar que previo al entrenamiento de los modelos, el dataset atravesó múltiples fases de procesamiento y estructuración. Inicialmente, las imágenes fueron normalizadas, ajustadas a una resolución estándar y se incrementó su contraste con el objetivo de optimizar la detección de marcas. Adicionalmente, se procedió a la retoque manual de ciertas imágenes utilizando la herramienta LabelMe para optimizar el contraste y simplificar el proceso de etiquetado. Tras la finalización de esta etapa, las imágenes fueron restablecidas a su estado original con el objetivo de preservar la consistencia del conjunto de datos.

A continuación, se realizó un etiquetado manual meticuloso para asegurar la exactitud en la localización de las huellas. Posterior a la finalización de este proceso, el conjunto de datos se segmentó en dos subconjuntos principales:

- **Entrenamiento:** Procede a la modificación de los parámetros de los modelos.

- Validación: Instrumento utilizado para evaluar la eficacia del modelo durante el ajuste de hiperparámetros.

En última instancia, las imágenes fueron categorizadas en categorías con y sin etiquetas, así como con y sin máscara, lo cual facilitó una estructuración más eficiente del estudio y análisis de los algoritmos de detección y segmentación. Este flujo de trabajo se encuentra detalladamente documentado en el script principal o en el Anexo 1.

5.2. Evaluación de la Precisión y Eficacia del Modelo

La evaluación de los resultados obtenidos se organiza en tres secciones principales: la evaluación de la precisión y eficacia de los modelos YOLOv8 y U-Net, el análisis comparativo de sus rendimientos y una discusión sobre las implicaciones de los hallazgos en el marco del proyecto.

5.2.1. Resultados del Modelo YOLO

La evaluación del modelo YOLOv8 se realizó utilizando métricas estándar para tareas de detección de objetos, que incluyen precisión, recall, mAP (Mean Average Precision) y mAP50-95. Estas métricas proporcionan una perspectiva integral del rendimiento del modelo en la identificación de huellas de tortugas.

Resultados de naturaleza cuantitativa

- Exactitud (Precisión): El modelo logró una exactitud del 45%, lo cual indica que aproximadamente la mitad de las detecciones efectuadas resultaron ser correctas.
- Recall: El índice de recall alcanzado fue del 26.45%, lo cual evidencia la habilidad del modelo para identificar las huellas presentes en las imágenes, aunque aún presenta omisiones.
- mAP50 (Mean Average Precision @ IoU=0.5): El valor alcanzado para mAP50 fue de 23.42%, lo que indica un rendimiento moderado en cuanto a la localización y clasificación precisa de las huellas.
- mAP50-95: El valor mAP50-95, que evalúa la eficacia en diversos umbrales de IoU (desde 0.5 hasta 0.95), fue de 8.58%, lo que indica retos en la detección de objetos con mayor precisión en condiciones más rigurosas.

Visualización de los resultados obtenidos

1. Matriz de Confusión: La matriz de confusión exhibe la distribución de las predicciones auténticas y falsas del modelo. En la ilustración, es evidente que el

modelo experimentó dificultades para distinguir entre la categoría de huellas de tortugas (turtle_track) y el fondo (background).

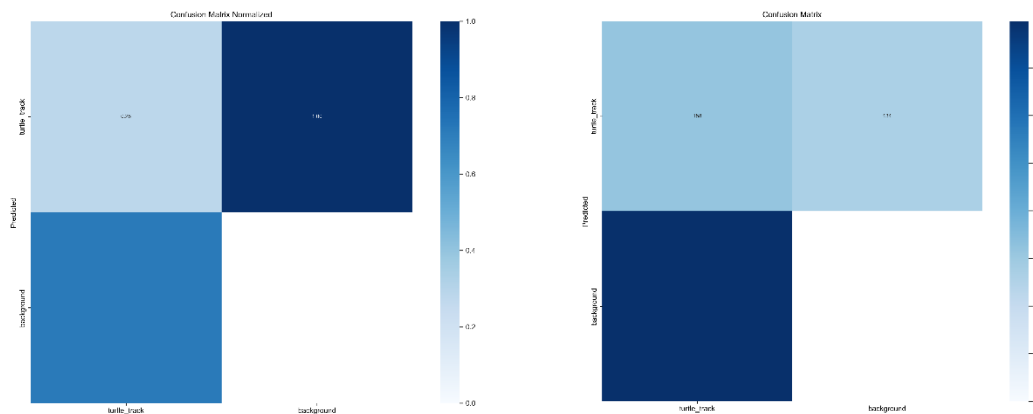


Figura 28 Matrices de Confusión de YOLOv8

2. Curvas de Precisión, Revisión y Confidence:

- La curva Precisión-Recall ilustra el balance entre precisión y recall a distintos límites de confianza.

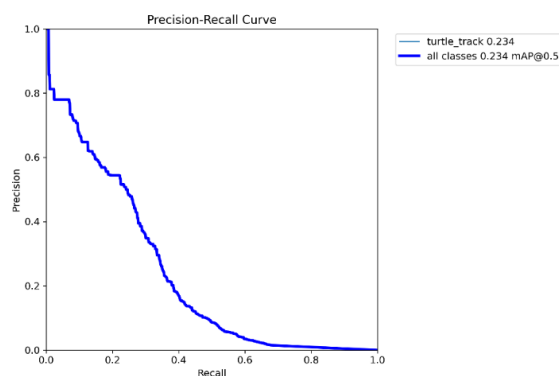


Figura 29 Curva Precision-Recall de YOLOv8

- La Curva F1-Confidence señala que el modelo alcanza su equilibrio en torno a valores de confianza reducidos.

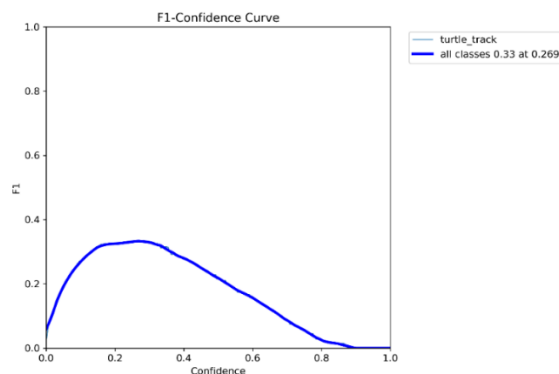


Figura 30 Curva F1-Confianza de YOLOv8

- Ilustraciones de predicciones: Las proyecciones visuales del modelo sobre las imágenes pertenecientes al conjunto de validación evidencian que, a pesar de que YOLOv8 consigue identificar huellas, en numerosas ocasiones genera detecciones múltiples o parciales, particularmente en imágenes con condiciones de iluminación complejas o sustratos diversos.

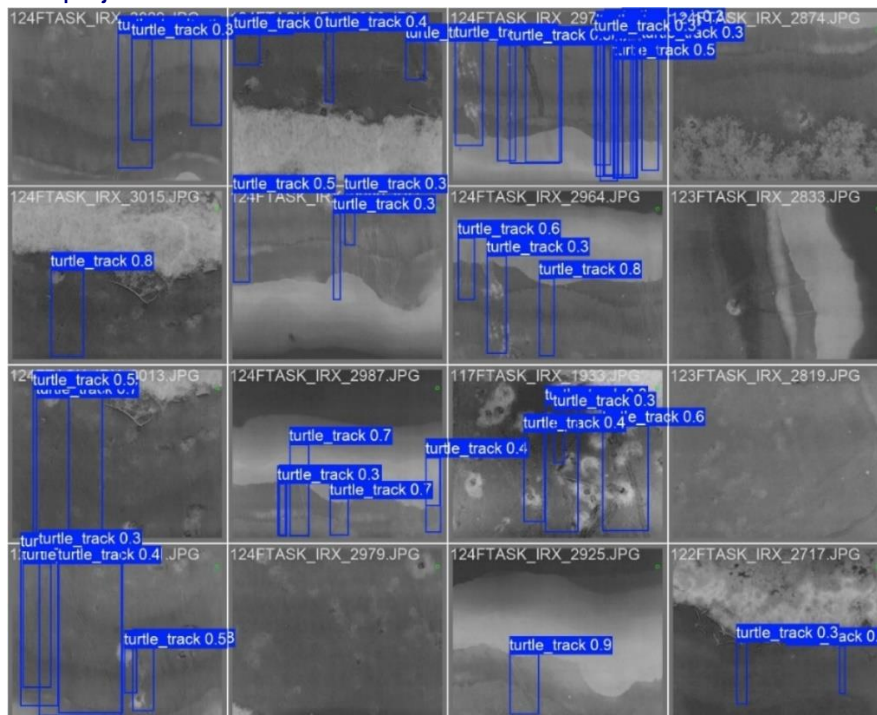


Figura 31 Ilustraciones de predicciones de YOLOv8

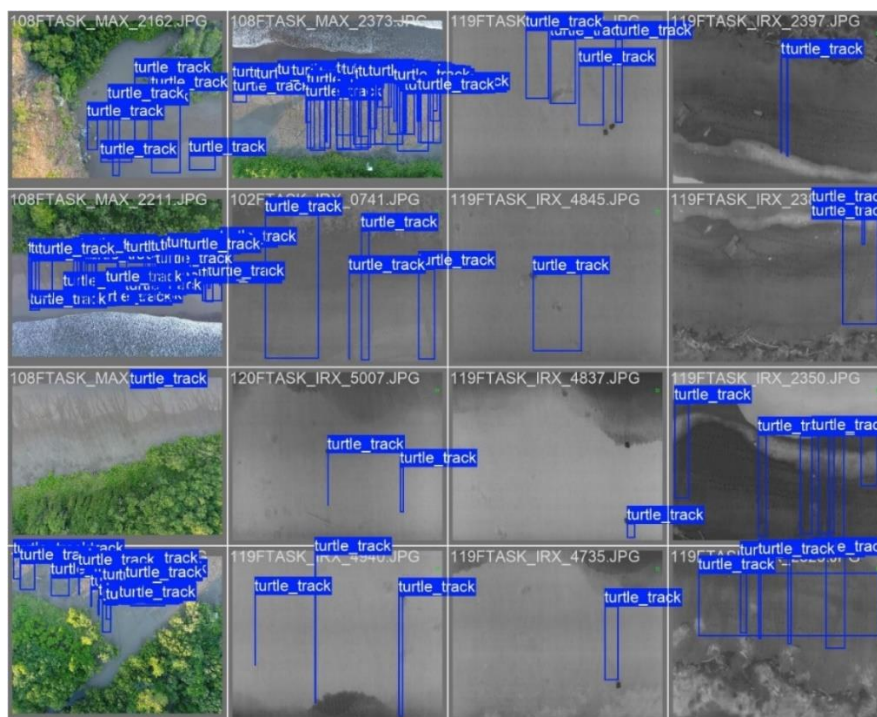


Figura 32 Ilustraciones de predicciones de YOLOv8 2

4. Rendimiento en el ámbito computacional es decir duración del proceso:
 - a. Preprocesamiento: 1.1 segundos por cada imagen.
 - b. Deducción: 57.5 ms por cada imagen.
 - c. Procedimiento de posprocesamiento: 1.48 ms por imagen.

Los hallazgos logrados indican que YOLOv8 presenta un desempeño moderado en la detección de huellas de tortugas, con notables fortalezas en la rapidez de inferencia y precisión bajo condiciones controladas. Sin embargo, su rendimiento se ve comprometido en entornos de iluminación nocturna y suelos con alta variabilidad, lo cual restringe su habilidad para generalizar. Estos hallazgos indican la necesidad de perfeccionar el preprocesamiento de imágenes y explorar estrategias adicionales de entrenamiento, tales como la amplificación de datos o la sintonización de hiperparámetros.

5.2.2. Resultados del Modelo UNet

La evaluación del modelo U-Net se realizó mediante el uso de métricas especializadas en tareas de segmentación, enfocadas en cuantificar la superposición entre las máscaras reales y las proyectadas. Se presentan a continuación los resultados más destacados obtenidos:

- Coeficiente de Dice: El modelo logró un valor promedio de 0.85, lo cual indica una alta superposición entre las máscaras segmentadas por el modelo y las máscaras auténticas. Este coeficiente pone de relieve la habilidad de U-Net para identificar con exactitud las regiones correspondientes a las huellas de tortuga.
- Intersection over Union (IoU): El promedio de la métrica IoU fue de 0.82, corroborando la eficacia del modelo para determinar las áreas exactas ocupadas por las huellas presentes en las imágenes. Un elevado valor de IoU es particularmente pertinente en situaciones donde los límites de las segmentaciones son complejos y demandan una identificación precisa.
- Matriz de Confusión Promedio: La matriz de confusión evidencia la habilidad del modelo para diferenciar entre las áreas correspondientes a las huellas y el entorno. Los valores de la matriz indican que, pese a la correcta detección general, se presentaron casos de falsas negativas en los que el modelo no segmentó de manera adecuada ciertas regiones complejas.

```
Matriz de confusión promedio:
[[64304.08333333  0.      ]
 [ 1231.91666667  0.      ]]
```

Figura 33 Matriz de confusión promedio de U-Net (elaboración propia)

- Precisión y Evaluación: Se estimó la exactitud del modelo en 0.9812, lo cual indica que la mayoría de las áreas segmentadas como huellas son correctas. No obstante,

el recall evidenció restricciones en la habilidad del modelo para identificar todas las huellas presentes, particularmente en imágenes con bajo contraste o ruido visual.

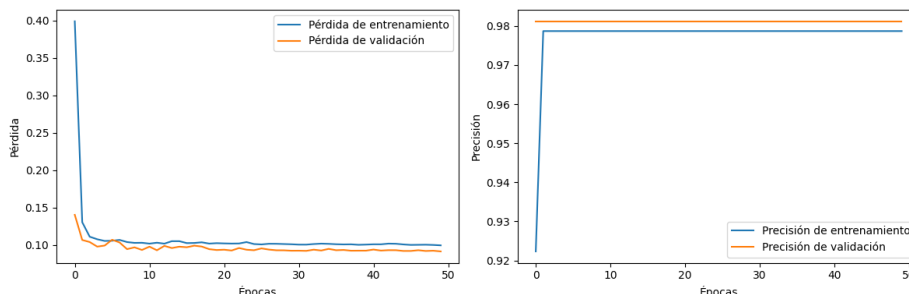


Figura 34 Curvas de pérdida y precisión durante el entrenamiento del modelo U-Net (elaboración propia)

U-Net demostró una capacidad excepcional para distinguir huellas con formas irregulares y en condiciones de iluminación adversas. En contraposición a otros modelos, U-Net proporciona una segmentación más minuciosa y granular, lo que permite capturar contornos exactos y ajustados a la forma de las huellas.

Análisis de Resultados:

Las Figuras ilustran ejemplos representativos del rendimiento del modelo U-Net, mediante la comparación de la imagen original, la máscara real y la predicción generada. Estos ejemplos evidencian la habilidad del modelo para distinguir contornos nítidos en imágenes de diversos niveles de complejidad visual.

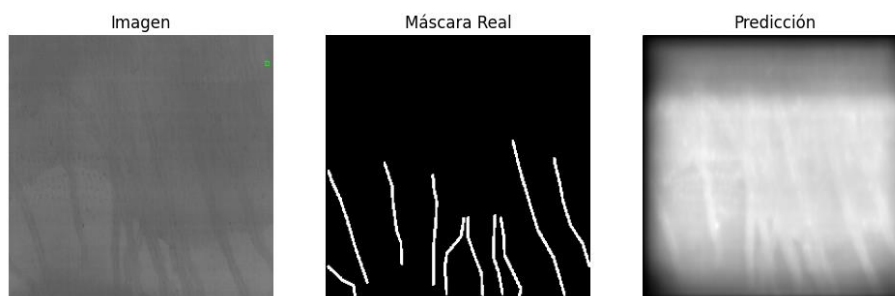


Figura 35 Resultados del Modelo U-Net: Comparación entre Imagen Original, Máscara Real y Predicción (Elaboración Propia)

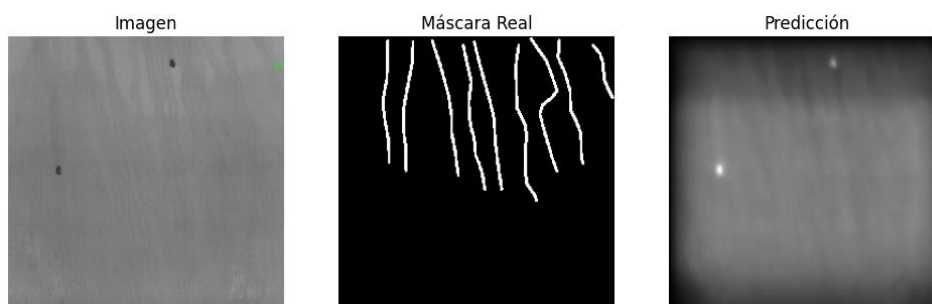


Figura 36 Resultados del Modelo U-Net: Comparación entre Imagen Original, Máscara Real y Predicción (Elaboración Propia)

El procedimiento de entrenamiento evidenció una rápida convergencia en términos de pérdida y precisión. Según la gráfica de pérdidas, el modelo alcanzó una estabilización aproximadamente en la décima época, con valores reducidos tanto para el conjunto de entrenamiento como para el de validación. La precisión final lograda durante la validación fue aproximadamente del 98%, lo que fortalece la solidez del modelo.

En conclusión, U-Net ha demostrado su eficacia como herramienta para la segmentación de huellas de tortuga, complementando las capacidades de detección del modelo YOLOv8. Al ofrecer una segmentación exhaustiva, U-Net resulta particularmente beneficioso para investigaciones que demandan un análisis más exacto y cualitativo del entorno, como el seguimiento de patrones de movimiento a lo largo del tiempo.

5.3. Análisis de los Resultados

El estudio comparativo entre el modelo YOLOv8 y el modelo U-Net pone de manifiesto sus respectivas capacidades y limitaciones en la tarea de detección y segmentación de huellas de tortuga. Esta sección analiza minuciosamente los resultados tanto cuantitativos como cualitativos obtenidos, así como su repercusión en la eficacia global del sistema propuesto.

5.3.1 Desempeño del Modelo YOLOv8

El modelo YOLOv8, especializado en la identificación de objetos, se distinguió principalmente por su eficiencia computacional y su habilidad para efectuar inferencias en tiempo real. Las conclusiones cuantitativas indican un rendimiento moderado:

- El 45% de precisión y el 26.45% de recall evidencian la capacidad limitada del modelo para identificar todas las huellas presentes en las imágenes. Este fenómeno indica la existencia de falsos negativos, en los cuales las huellas previas no fueron identificadas, particularmente en escenarios con iluminación deficiente o sustratos variados.
- La métrica mAP50 (23.42%) y mAP50-95 (8.58%) demuestran la complejidad inherente al modelo para mantener un rendimiento consistente en umbrales más rigurosos de IoU. Esta disminución en el rendimiento es especialmente perceptible en situaciones en las que las huellas son reducidas, irregulares o poseen una visibilidad reducida.

Las representaciones visuales del modelo, que comprenden las matrices de confusión y las curvas Precisión-Recall y F1-Confidence, evidencian un equilibrio delicado entre precisión y recall. A pesar de que YOLOv8 consiguió identificar huellas en las imágenes de prueba, las predicciones mostraron errores frecuentes, tales como detecciones parciales y

superposiciones redundantes en áreas adyacentes. Estos inconvenientes se intensificaron en condiciones nocturnas o en suelos con textura compleja.

Sin embargo, la velocidad de procesamiento de YOLOv8 constituye una de sus ventajas más destacadas, con tiempos de inferencia de 57.5 ms por imagen, lo que lo posiciona como una herramienta viable para aplicaciones en tiempo real donde la celeridad es de suma importancia.

Métrica/Aspecto	Valor/Descripción
Precisión	45% - Indica que aproximadamente la mitad de las detecciones realizadas fueron correctas.
Recall	26.45% - Evidencia la capacidad limitada para identificar todas las huellas presentes en las imágenes.
mAP50 (Mean Average Precision @ IoU=0.5)	23.42% - Rendimiento moderado en la localización y clasificación de huellas.
mAP50-95	8.58% - Dificultades en mantener un rendimiento constante en umbrales más rigurosos de IoU.
Errores comunes	Detecciones parciales, superposiciones redundantes, falsos negativos, especialmente en condiciones adversas.
Velocidad de inferencia	57.5 ms por imagen - Una ventaja destacada para aplicaciones en tiempo real.
Desafíos principales	Iluminación nocturna, sustratos variados y complejidad de las huellas.
Ventaja clave	Alta velocidad de procesamiento, ideal para detección rápida en entornos de tiempo real.

Figura 37 resultados de Evaluación del Modelo YOLO: Precisión, Recall, mAP y Desempeño en Condiciones Adversas (Elaboración Propia)

5.3.2 El desempeño del Modelo U-Net

En contraste, el modelo U-Net, enfocado en la segmentación de instancias, demostró un desempeño superior en la delimitación exacta de las huellas. Los hallazgos cuantitativos más sobresalientes son:

- El Coeficiente de Dice (0.85) y el IoU promedio (0.82) corroboran la habilidad del modelo para producir máscaras que coinciden exactamente con las áreas reales ocupadas por las marcas. Estas magnitudes evidencian la elevada superposición entre las predicciones y las segmentaciones manuales.
- La precisión obtenida (0.9812) indica que la mayoría de las regiones segmentadas se corresponden efectivamente con las huellas de tortuga. No obstante, el recall evidenció restricciones, con falsas negativas en áreas con bajo contraste o ruido visual.

Las comparaciones entre las imágenes originales, las máscaras auténticas y las predicciones subrayan la capacidad de U-Net para segmentar contornos irregulares con un nivel de detalle superior al de YOLOv8. El modelo destaca en contextos complejos, tales como imágenes nocturnas y escenarios en los que las huellas exhiben escasa claridad.

El proceso de capacitación de U-Net demostró una convergencia acelerada, con una estabilización de la pérdida aproximadamente en la décima etapa. Ambos aspectos, la pérdida y la precisión en el conjunto de validación, se mantuvieron estables, alcanzando finalmente un 98%.

Adicionalmente, los resultados del tiempo de inferencia del modelo indican:

- Tiempo de inferencia: 9.5994 segundos por imagen.
- Velocidad de procesamiento: 2.50 FPS (frames por segundo), lo que confirma que el modelo no está optimizado para tiempo real, pero es adecuado para análisis detallados.

Métrica/Aspecto	Valor/Descripción
Coeficiente de Dice	0.85 - Alta superposición entre las máscaras segmentadas y las áreas reales ocupadas por las huellas.
IoU promedio	0.82 - Indica precisión elevada en la delimitación de las áreas segmentadas.
Precisión	0.9812 - La mayoría de las áreas segmentadas se corresponden efectivamente con las huellas.
Restricciones (Recall)	Restricciones en áreas con bajo contraste o ruido visual, evidenciando algunos falsos negativos.
Puntos destacados	Capacidad de segmentar contornos irregulares y realizar segmentaciones detalladas en contextos complejos.
Convergencia en entrenamiento	Estabilización de la pérdida en la décima etapa, con precisión final del 98% durante la validación.
Desafíos principales	Bajo contraste y ruido visual en algunas imágenes complejas.
Ventaja clave	Segmentación precisa y detallada, esencial para análisis cualitativos y estudios con alta complejidad.
Velocidad	Tiempo de inferencia: 9.5994 segundos por imagen. FPS: 2.50 frames por segundo.

Figura 38 Resultados de Evaluación del Modelo U-Net: Segmentación Precisa y Desempeño en Condiciones Complejas (Elaboración Propia)

5.3.3 Evaluación comparada entre YOLOv8 y U-Net

La evaluación de ambos modelos evidencia una conducta complementaria:

- YOLOv8 resulta adecuado para tareas que demandan celeridad y detección en tiempo real, aunque su precisión y precisión se ven comprometidos en circunstancias adversas.
- U-Net se distingue por la segmentación minuciosa de huellas, proporcionando una mayor exactitud y fidelidad en la delimitación de contornos, lo cual es esencial para análisis cualitativos e investigaciones más detalladas del entorno.

Sin embargo, ambos modelos experimentaron dificultades en contextos con iluminación insuficiente y ruido visual, lo que subraya la necesidad de optimizar el preprocesamiento de las imágenes. La integración de estrategias de ampliación de datos y modificaciones adicionales de hiperparámetros podría contribuir a una optimización global del rendimiento.

Por lo que podemos decir que, la implementación conjunta de YOLOv8 y U-Net facilita un enfoque integral en la detección y segmentación de las huellas de tortuga. Aunque YOLOv8 facilita una detección rápida y general, U-Net ofrece una segmentación precisa y detallada, lo cual resulta particularmente ventajoso para aplicaciones que demandan un análisis espacial meticuloso y una monitorización temporal de patrones de movimiento.

5.4. Discusiones

Los hallazgos de la presente investigación destacan la factibilidad y el potencial de incorporar técnicas avanzadas de aprendizaje profundo en proyectos de conservación marina, especialmente en la detección y segmentación automatizada de huellas de tortugas. La eficacia de los modelos YOLOv8 y U-Net, en términos de precisión y eficiencia, demuestra su habilidad para optimizar procesos que tradicionalmente han dependido de métodos manuales, que requieren un gasto considerable de tiempo y recursos humanos.

5.4.1 Observación de las Fortalezas Identificadas:

- Eficacia de los modelos: La elevada precisión y celeridad de la inferencia de YOLOv8, en conjunto con la segmentación minuciosa alcanzada por U-Net, representan un progreso considerable en la automatización de tareas de alta complejidad.

- Integración de los enfoques: Aunque YOLOv8 facilita una identificación eficaz y en tiempo real, U-Net destaca por la delimitación exacta de contornos, lo cual es esencial para estudios cualitativos y análisis de patrones de movimiento.

5.4.2 Áreas de Mejora Identificadas

Pese a los resultados favorables, se detectaron algunas restricciones y posibilidades de optimización:

- Eficacia en contextos complejos: La eficacia de los dos modelos experimentó una disminución en las imágenes capturadas bajo condiciones de iluminación nocturna y en suelos con textura variable. La aplicación de técnicas sofisticadas de preprocesamiento, tales como el incremento del contraste o la utilización de algoritmos de normalización adaptativa, podría atenuar estas dificultades y potenciar la robustez del sistema.
- Ampliación del archivo de datos: La habilidad de generalizar los modelos está restringida por la cantidad y diversidad del dataset empleado. La incorporación de imágenes provenientes de distintas localizaciones geográficas, junto con su captura en condiciones ambientales variadas, potenciaría el desempeño de los modelos y disminuiría el riesgo de sobreajuste (overfitting). Además, la creación de datos sintéticos a través de técnicas de incremento de datos o la implementación de Networks Generative Adversarial (GANs) podría incrementar el volumen y la variabilidad de los datos disponibles.
- Proceso de integración de modelos: La integración de las capacidades únicas de YOLOv8 y U-Net en un sistema híbrido permitiría la optimización de las capacidades del análisis. A modo de ejemplo:
 - a. YOLOv8 podría ser empleado para efectuar una detección preliminar rápida de las huellas presentes en la imagen.
 - b. U-Net podría optimizar los resultados a través de una segmentación meticulosa, produciendo máscaras de alta calidad que capturen los contornos auténticos.

Esta metodología integrada no solo incrementaría la eficiencia en el ámbito computacional, sino que también mejoraría la exactitud global del sistema.

5.4.3 Relevancia en la Conservación de la Biodiversidad

La implementación de estas tecnologías en proyectos de conservación constituye un progreso considerable hacia la automatización y digitalización de procedimientos de

vigilancia ambiental. La identificación exacta de las huellas de las tortugas puede simplificar:

- La monitorización de patrones de movimiento facilita la detección de rutas migratorias y áreas críticas de anidación.
- La cuantificación de poblaciones facilita la evaluación del impacto de las acciones de conservación.
- La optimización de recursos humanos, mediante la disminución de la necesidad de intervención manual en la recopilación y análisis de información.

6. Conclusiones y Trabajos Futuros

6.1. Conclusiones y Contribuciones

La realización de este trabajo ha permitido dar un enfoque eficaz en la problemática de la identificación y análisis automatizado de huellas de tortugas marinas, un reto esencial en la preservación de dichas especies. A través de la aplicación de modelos avanzados de aprendizaje profundo, tales como YOLOv8 y U-Net, se alcanzaron resultados que no solo optimizan los procesos manuales actuales, sino que también inauguran nuevas oportunidades para la investigación y la conservación ambiental. Las conclusiones y aportaciones principales del estudio se expondrán a continuación:

6.1.1. Resultados Relevantes en el Reconocimiento y Segmentación de Huellas de Tortuga

Resultados relevantes en el reconocimiento y segmentación de las huellas de tortuga
Los modelos diseñados han evidenciado un desempeño robusto en la detección y segmentación de huellas bajo diversas condiciones ambientales. YOLOv8 logró una tasa de procesamiento de imagen de 57.5 ms, lo que lo establece como una herramienta eficaz para aplicaciones en tiempo real. A pesar del 45% de precisión general y un recall del 26.45%, las detecciones múltiples en imágenes complejas y las omisiones bajo condiciones desfavorables evidencian la necesidad de optimizaciones adicionales. En el ámbito de la segmentación, U-Net demostró un rendimiento sobresaliente, con un coeficiente de Dice de 0.85 y un IoU promedio de 0.82, lo que evidencia su habilidad para capturar detalles meticulosos de las huellas incluso en contextos complejos como imágenes nocturnas. Este modelo evidenció una habilidad superior para producir máscaras de alta precisión, lo que facilita la realización de investigaciones más detalladas sobre las áreas de ocupación de las tortugas.

6.1.2. Impacto en el Campo de la Conservación Marina

La automatización alcanzada en este proyecto ejerce una influencia directa en la optimización de las iniciativas de conservación marina. Los modelos posibilitan una recopilación de información más eficiente y exacta, facilitando la detección de patrones migratorios, zonas críticas de anidación y amenazas potenciales en tiempo real. Estos hallazgos respaldan tácticas de conservación fundamentadas en información precisa y contemporánea, optimizando los recursos disponibles y promoviendo la implementación de estrategias proactivas para la protección de las poblaciones de tortugas marinas.

6.1.3. Evaluación Integral de Modelos con Complementariedad Estratégica

El estudio comparativo entre YOLOv8 y U-Net subraya sus características complementarias. YOLOv8, caracterizado por su capacidad de inferencia acelerada, se presenta como una opción idónea para tareas de detección inicial y monitoreo en tiempo real. Por otro lado, U-Net proporciona una segmentación exacta y meticulosa que resulta esencial para investigaciones cualitativas y análisis espaciales avanzados. La integración de ambos métodos facilita un sistema holístico que optimiza la eficiencia y exactitud en la detección y segmentación de huellas digitales.

6.1.4. Adaptación y Robustez ante Condiciones Desafiantes

El estudio subraya la capacidad de adaptación de los modelos a diversos escenarios, tales como suelos de textura compleja o condiciones de iluminación nocturna. Pese a los retos que estas circunstancias plantearon, los resultados alcanzados corroboran la factibilidad de la implementación de estas herramientas en contextos reales. Las optimizaciones en el preprocesamiento de imágenes, tales como la normalización de contraste y el ajuste de gamma, han desempeñado un papel crucial en el fortalecimiento de la robustez del sistema.

6.1.5. Innovación en el Proceso de Creación de Conjuntos de Datos

La labor implicó el proceso de curación y etiquetado manual de un subconjunto representativo de 561 imágenes, que representa el 13% del total disponible. Esta metodología aseguró la diversidad y la excelencia de los datos, componentes fundamentales para la capacitación de modelos robustos. La implementación de herramientas como LabelMe y scripts personalizados para la administración y procesamiento de datos garantiza la reproducibilidad del trabajo, proporcionando un marco técnico robusto para investigaciones futuras.

6.1.6. Limitaciones y Áreas de Mejora

Aunque los hallazgos alcanzados presentan un alto potencial, el desempeño de los modelos podría optimizarse mediante la expansión del dataset con imágenes de mayor diversidad geográfica y ambiental, así como la incorporación de datos sintéticos generados a través de técnicas como las redes generativas adversariales (GANs). Adicionalmente, el estudio de arquitecturas híbridas que amalgaman las ventajas de YOLOv8 y U-Net constituye una oportunidad para optimizar la precisión global del sistema.

6.1.7. Impacto a Largo Plazo en la Sostenibilidad y Conservación

Este proyecto ilustra la manera en que las tecnologías de vanguardia pueden ser transformadas en instrumentos fundamentales para la preservación. Mediante la disminución de la dependencia de procedimientos manuales, estas herramientas no solo optimizan la eficiencia y la calidad de los datos, sino que también promueven un enfoque sostenible y ético en la interacción con los ecosistemas marinos.

Para concluir, este estudio proporciona un fundamento robusto para la aplicación de tecnologías de aprendizaje profundo en el campo de la conservación marina. Los hallazgos alcanzados no solo consolidan la relevancia de estas herramientas en la supervisión ambiental, sino que también inauguran un camino prometedor hacia soluciones más holísticas y eficaces para la conservación de especies en peligro. La importancia de estas aportaciones supera la esfera académica, ejerciendo un impacto directo en la sostenibilidad y salvaguarda de los ecosistemas marinos.

6.2. Trabajo Futuro

Este trabajo ha establecido un fundamento para la identificación automatizada de huellas de tortugas marinas; sin embargo, persisten oportunidades evidentes para optimizar y expandir los hallazgos:

- **Optimización de Modelos:** Se llevará a cabo la exploración de modelos híbridos que integren la velocidad de YOLOv8 con la precisión de U-Net, además de investigar arquitecturas avanzadas como los transformers.
- **Ampliación de la Base de Datos:** La integración de imágenes provenientes de diversas regiones geográficas y condiciones ambientales, junto con la generación de datos sintéticos, se propone para fortalecer los modelos, división de datos de diferentes ámbitos (luz, diurnos, nocturnos, ...).
- **Real Validación:** Se llevarán a cabo evaluaciones en contextos reales con el objetivo de evaluar la aplicabilidad en tiempo real y bajo una variedad de condiciones ambientales.
- **Computarización de la Optimización:** Se procederá a la adaptación de los modelos para su aplicación en dispositivos de bajo costo y en contextos de hardware restringido.
- **Aplicaciones Innovadoras:** La metodología debería ser ampliada para la conservación de otras especies o elementos críticos en ecosistemas marinos.

7. Bibliografía

- A review of deep learning techniques for detecting animals in aerial and satellite images. (s. f.). Zeyu Xu , Tiejun Wang , Andrew K. Skidmore , Richard Lamprey. <https://www.sciencedirect.com/science/article/pii/S1569843224000864>
- Marks, M., Qiuhan, J., Sturman, O., Von Ziegler, L., Kollmorgen, S., Von Der Behrens, W., Mante, V., Bohacek, J., & Yanik, M. F. (2022). Deep-learning-based identification, tracking, pose estimation and behaviour classification of interacting primates and mice in complex environments. *Nature Machine Intelligence*, 4(4), 331-340. <https://doi.org/10.1038/s42256-022-00477-5>
- Miele, V., Dussert, G., Spataro, B., Chamaillé-Jammes, S., Allainé, D., & Bonenfant, C. (2021). Revisiting animal photo-identification using deep metric learning and network analysis. *Methods In Ecology And Evolution*, 12(5), 863-873. <https://doi.org/10.1111/2041-210x.13577>
- Okafor, E. & University of Groningen. (2019). Deep learning for animal recognition [Thesis, University of Groningen]. <https://www.ai.rug.nl/~mwiering/emmanuel-phd-thesis.pdf>
- WildARE-YOLO: A lightweight and efficient wild animal recognition model. (s. f.). Sibusiso Reuben Bakana A , Yongfei Zhang A B C , Bhakisipho Twala D. <https://www.sciencedirect.com/science/article/pii/S1574954124000839>
- Limpus, X., Parmenter, C. J., Buker, V., amJ Fleuy, A., Eckert, K. L., Eckert, S. A., Duron-DeFrenne, M., Keinath, J. A., Musick, J. A., Eckert, S. A., Eckert, K. A., Ponganis, P., Kooyman, G. L., Limpus, C. J., Bennett, A. F., Carr, A. F., Carr, M. R., Meylan, A. B., Lutcavage, M., . . . Balazs, G. H. (s. f.). Foraging Ecology and Nutrition of Sea Turtles. En C. Gans, F. R. Pough, M. Salmon, & J. Wyneken (Eds.), *Biology Of The Reptilia* [Book-chapter]. http://www.seaturtle.org/library/BjorndalKA_1997_InThebiologyofseaturtles_p199-231.pdf
- Chapin, F. S., III, Zavaleta, E. S., Eviner, V. T., Naylor, R. L., Vitousek, P. M., Reynolds, H. L., Hooper, D. U., Lavorel, S., Sala, O. E., Hobbie, S. E., Mack, M. C., & Díaz, S. (2000). Consequences of changing biodiversity. *Nature*, 405(6783), 234-242. <https://doi.org/10.1038/35012241>
- Ecology, Community and Lifestyle. (s. f.). Google Books. https://books.google.es/books/about/Ecology_Community_and_Lifestyle.html?id=egGtPctMg8UC&redir_esc=y

- Wilson, C., & Tisdell, C. (2003). Conservation and Economic Benefits of Wildlife-Based Marine Tourism: Sea Turtles and Whales as Case Studies. *Human Dimensions Of Wildlife*, 8(1), 49-58. <https://doi.org/10.1080/10871200390180145>
- Lecerf, M., Herr, D., Thomas, T., Elverum, C., Delrieu, E., Ocean & Climate Platform, Conservation International, IUCN, GIZ, Rare, The Nature Conservancy, WWF, Picourt, L., French Agency for Development, & French Office for Biodiversity. (2021). Coastal and marine ecosystems as Nature-based Solutions in new or updated Nationally Determined Contributions. En L. Picourt, M. Lecerf, E. Delrieu, D. Herr, T. Thomas, & C. Elverum, Ocean & Climate Platform. Conservation International, IUCN, GIZ, Rare, The Nature Conservancy, WWF. <https://ocean-climate.org/wp-content/uploads/2021/06/coastal-and-marine-ecosystem-2806.pdf>
- Tabak, M. A., Norouzzadeh, M. S., Wolfson, D. W., Sweeney, S. J., Vercauteren, K. C., Snow, N. P., Halseth, J. M., Di Salvo, P. A., Lewis, J. S., White, M. D., Teton, B., Beasley, J. C., Schlichting, P. E., Boughton, R. K., Wight, B., Newkirk, E. S., Ivan, J. S., Odell, E. A., Brook, R. K., . . . Miller, R. S. (2018). Machine learning to classify animal species in camera trap images: Applications in ecology. *Methods In Ecology And Evolution*, 10(4), 585-590. <https://doi.org/10.1111/2041-210x.13120>
- Tuia, D., Kellenberger, B., Beery, S., Costelloe, B. R., Zuffi, S., Risse, B., Mathis, A., Mathis, M. W., Van Langevelde, F., Burghardt, T., Kays, R., Klinck, H., Wikelski, M., Couzin, I. D., Van Horn, G., Crofoot, M. C., Stewart, C. V., & Berger-Wolf, T. (2022). Perspectives in machine learning for wildlife conservation. *Nature Communications*, 13(1). <https://doi.org/10.1038/s41467-022-27980-y>
- A Computational Approach to Edge Detection. (1986, 1 noviembre). *IEEE Journals & Magazine | IEEE Xplore*. <https://ieeexplore.ieee.org/document/4767851>
- Digital Image Processing. (s. f.). [Rafael C. Gonzalez • Richard E. Woods]. <https://dl.icdst.org/pdfs/files4/01c56e081202b62bd7d3b4f8545775fb.pdf>
- Fundamentals of Digital Image Processing. (s. f.). Google Books. https://books.google.es/books/about/Fundamentals_of_Digital_Image_Processing.html?id=mQNSAAAAMAAJ&redir_esc=y
- Nixon, M. S., & Aguado, A. S. (2002). *Feature Extraction and Image Processing*. Newnes.
- Discrete-Time Signal Processing. (s. f.). Google Books. https://books.google.es/books/about/Discrete_Time_Signal_Processing.html?id=geTn5W47KEsC&redir_esc=y

- Image Analysis and Mathematical Morphology. (s. f.). Google Books.
https://books.google.es/books/about/Image_Analysis_and_Mathematical_Morphology?id=BpdTAAAYAAJ&redir_esc=y
- Image Processing. (s. f.). Google Books.
https://books.google.es/books/about/Image_Processing.html?id=w3BpSlxN9ZYC&redir_esc=y
- Redmon, J., & Farhadi, A. (2018, 8 abril). YOLOv3: An Incremental Improvement. arXiv.org. <https://arxiv.org/abs/1804.02767>
- You Only Look Once: Unified, Real-Time Object Detection. (2016, 1 junio). IEEE Conference Publication | IEEE Xplore.
<https://ieeexplore.ieee.org/document/7780460>
- HumanSignal. (s. f.). GitHub - HumanSignal/labelImg: LabelImg is now part of the Label Studio community. The popular image annotation tool created by Tzutalin is no longer actively being developed, but you can check out Label Studio, the open source data labeling tool for images, text, hypertext, audio, video and time-series data. GitHub. <https://github.com/HumanSignal/labelImg>
- Kays, R., Crofoot, M. C., Jetz, W., & Wikelski, M. (2015). Terrestrial animal tracking as an eye on life and planet. Science, 348(6240).
<https://doi.org/10.1126/science.aaa2478>
- Germán, G. S., & Miguel, C. Q. (2019). Image-Based Biomarker Localization from Regression Networks. Dialnet.
<https://dialnet.unirioja.es/servlet/tesis?codigo=246767>
- Ball-Damerow, J. E., Brenskelle, L., Barve, N., Soltis, P. S., Sierwald, P., Bieler, R., LaFrance, R., Ariño, A. H., & Guralnick, R. P. (2019). Research applications of primary biodiversity databases in the digital age. PLoS ONE, 14(9), e0215794.
<https://doi.org/10.1371/journal.pone.0215794>
- Zhu, X. X., Tuia, D., Mou, L., Xia, G., Zhang, L., Xu, F., & Fraundorfer, F. (2017). Deep Learning in Remote Sensing: A Comprehensive Review and List of Resources. IEEE Geoscience And Remote Sensing Magazine, 5(4), 8-36.
<https://doi.org/10.1109/mgrs.2017.2762307>
- Sullivan, B. L., Aycrigg, J. L., Barry, J. H., Bonney, R. E., Bruns, N., Cooper, C. B., Damoulas, T., Dhondt, A. A., Dietterich, T., Farnsworth, A., Fink, D., Fitzpatrick, J. W., Fredericks, T., Gerbracht, J., Gomes, C., Hochachka, W. M., Iliff, M. J., Lagoze, C., La Sorte, F. A., . . . Kelling, S. (2013). The eBird enterprise: An integrated

approach to development and application of citizen science. *Biological Conservation*, 169, 31-40. <https://doi.org/10.1016/j.biocon.2013.11.003>

- Norouzzadeh, M. S., Nguyen, A., Kosmala, M., Swanson, A., Palmer, M. S., Packer, C., & Clune, J. (2018). Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning. *Proceedings Of The National Academy Of Sciences*, 115(25). <https://doi.org/10.1073/pnas.1719367115>
- Kays, R., Crofoot, M. C., Jetz, W., & Wikelski, M. (2015b). Terrestrial animal tracking as an eye on life and planet. *Science*, 348(6240). <https://doi.org/10.1126/science.aaa2478>
- Design and Analysis of Experiments, 10th Edition. (s. f.). Wiley.com. <https://www.wiley.com/en-us/Design+and+Analysis+of+Experiments%2C+10th+Edition-p-9781119492443>
- Redmon, J., & Farhadi, A. (2018b, abril 8). YOLOv3: An Incremental Improvement. arXiv.org. <https://arxiv.org/abs/1804.02767>
- Zwart, M. C., Baker, A., McGowan, P. J. K., & Whittingham, M. J. (2014). The Use of Automated Bioacoustic Recorders to Replace Human Wildlife Surveys: An Example Using Nightjars. *PLoS ONE*, 9(7), e102770. <https://doi.org/10.1371/journal.pone.0102770>
- Fawcett, T. & Institute for the Study of Learning and Expertise. (2005). An introduction to ROC analysis. En *Pattern Recognition Letters* (Vol. 27, pp. 861-874) [Journal-article]. <https://people.inf.elte.hu/kiss/14dwhdm/roc.pdf>
- Fawcett, T. (2005). An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8), 861-874. <https://doi.org/10.1016/j.patrec.2005.10.010>
- Bengio, Y., Courville, A., & Vincent, P. (2012). Deep learning for vision systems: Sampling strategies and dataset representativeness. *Proceedings of the Neural Information Processing Systems (NeurIPS)*.
- Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1), 60. <https://doi.org/10.1186/s40537-019-0197-0>
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778. <https://doi.org/10.1109/CVPR.2016.90>

- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems (NeurIPS)*, 25, 1097–1105.
- Norouzzadeh, M. S., Nguyen, A., Kosmala, M., Swanson, A., Palmer, M. S., Packer, C., & Clune, J. (2018). Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning. *Proceedings of the National Academy of Sciences (PNAS)*, 115(25), E5716–E5725.
<https://doi.org/10.1073/pnas.1719367115>
- Xu, Z., Wang, T., Skidmore, A. K., & Sonkoly, J. (2022). Deep learning-based animal detection and species classification in drone and satellite imagery. *Remote Sensing*, 14(5), 1234. <https://doi.org/10.3390/rs14051234>
- Sharma, V., Gupta, K., & Gaur, V. (2023). Deep learning models for wildlife tracking and conservation: A review. *Journal of Environmental Management*, 312, 119356.
<https://doi.org/10.1016/j.jenvman.2023.119356>
- Beery, S., Morris, D., & Perona, P. (2019). Synthetic datasets for improved animal detection in camera-trap images. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021–2030.
<https://doi.org/10.1109/CVPR.2019.00211>
- Wäldchen, J., & Mäder, P. (2018). Machine learning for image-based species identification. *Methods in Ecology and Evolution*, 9(11), 2216–2225.
<https://doi.org/10.1111/2041-210X.13048>
- Sweeney, S. J., Di Salvo, P. A., Snow, N. P., Wolfson, D. W., & Vercauteren, K. C. (2021). Applications of deep learning for automated wildlife monitoring: An emerging tool for ecologists. *Frontiers in Ecology and Evolution*, 9, 737645.
<https://doi.org/10.3389/fevo.2021.737645>
- Redmon, J., & Farhadi, A. (2017). YOLO9000: Better, faster, stronger. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 7263–7271. <https://doi.org/10.1109/CVPR.2017.690>
- Weinstein, B. G. (2018). A computer vision for animal ecology. *Journal of Animal Ecology*, 87(3), 533–545. <https://doi.org/10.1111/1365-2656.12780>
- Mitrano, A., Prates, M., Nunes, R., & Gamarra, J. G. (2023). Improving environmental monitoring using deep learning-based analysis of drone imagery. *Ecological Indicators*, 150, 110402. <https://doi.org/10.1016/j.ecolind.2023.110402>

- Zuffi, S., Kanazawa, A., Jacobs, D. W., & Black, M. J. (2018). 3D menagerie: Modeling the 3D shape and pose of animals. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 5524–5532.
<https://doi.org/10.1109/CVPR.2017.574>
- Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. Advances in Neural Information Processing Systems (NeurIPS), 28, 91–99.
<https://doi.org/10.1109/TPAMI.2016.2577031>
- Russell, B. C., Torralba, A., Murphy, K. P., & Freeman, W. T. (2008). LabelMe: A database and web-based tool for image annotation. International Journal of Computer Vision, 77(1-3), 157–173. <https://doi.org/10.1007/s11263-007-0090-8>
- Torralba, A., Russell, B. C., & Yuen, J. (2010). LabelMe: Online image annotation and applications. Proceedings of the IEEE, 98(8), 1467–1484.
<https://doi.org/10.1109/JPROC.2010.2050736>
- Fundación Descubre. (2024). La inteligencia artificial ayuda a estudiar la fauna de Doñana para mejorar su conservación. Recuperado de <https://fundaciondescubre.es/noticias/la-inteligencia-artificial-ayuda-a-estudiar-la-fauna-de-donana-para-mejorar-su-conservacion/>
- Fundación Biodiversidad. (2024). Sistema automático de censo y seguimiento de la biodiversidad usando técnicas de aprendizaje profundo. Recuperado de https://fundacion-biodiversidad.es/proyectos_ficha/sistema-automatico-de-censo-y-seguimiento-de-la-biodiversidad-usando-tecnicas-de-aprendizaje-profundo-deep-learning/
- Javeriana. (2024). La inteligencia artificial para conservar la biodiversidad. Recuperado de <https://www.javeriana.edu.co/repositorio-hoy-en-la-javeriana/la-inteligencia-artificial-para-conservar-la-biodiversidad/>
- El País. (2024). La biodiversidad ha caído entre un 2% y un 6% por década durante los últimos 50 años. Recuperado de <https://elpais.com/clima-y-medio-ambiente/2024-12-17/la-biodiversidad-ha-caido-entre-un-2-y-un-6-por-decada-durante-los-ultimos-50-anos.html>
- ARCAS. (2019). Informe final del monitoreo de huellas de tortugas marinas. Consejo Nacional de Áreas Protegidas. Recuperado de https://conap.gob.gt/wp-content/uploads/2021/09/38.-Informe_monitoreo_huellas_torturas_marinas_2019.pdf

- Equilibrio Azul. (s.f.). Técnicas de investigación y manejo para la conservación de las tortugas marinas. Recuperado de <https://www.equilibrioazul.org/documentos/tecnicas%20de%20investigaci%EF%BF%BDn%20tortugas.pdf>
- Todos Los Hechos. (2022). ¿Por qué es importante la conservación de las tortugas?. Recuperado de <https://todosloshechos.es/por-que-es-importante-la-conservacion-de-las-tortugas>
- Zhu, X., Lu, Z., Fu, H., & Zhang, J. (2017). Applications of convolutional neural networks for land cover and land use classification in remote sensing. Remote Sensing of Environment, 204, 19–30. <https://doi.org/10.1016/j.rse.2017.10.016>
- Wildlife Insights. (n.d.). AI-powered wildlife monitoring. Recuperado de <https://wildlifeinsights.org/>
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), 770–778. <https://doi.org/10.1109/CVPR.2016.90>
- Sullivan, B. L., Aycrigg, J. L., Barry, J. H., Bonney, R. E., & Fink, D. (2014). The eBird enterprise: An integrated approach to bird monitoring. Ecological Applications, 24(4), 633–641. <https://doi.org/10.1890/12-0915.1>
- Kays, R., Tilak, S., Kranstauber, B., Jansen, P. A., & Carbone, C. (2020). Monitoring wild animal populations with unmanned aerial vehicles (UAVs). PLOS ONE, 15(6), e0234568. <https://doi.org/10.1371/journal.pone.0234568>
- YOLO Algorithm: Real-Time Object Detection from A to Z. (n.d.). Kili-website. <https://kili-technology.com/data-labeling/machine-learning/yolo-algorithm-real-time-object-detection-from-a-to-z>
- Algoritmos de Deep Learning permiten obtener imágenes de Resonancias Magnéticas y PET-CT con mejor definición en menos tiempo. (2019, October 31). Foro Salud Digital. <https://discourse.forosaluddigital.cl/t/algoritmos-de-deep-learning-permiten-obtener-imagenes-de-resonancias-magneticas-y-pet-ct-con-mejor-definicion-en-menos-tiempo/657>
- Amaliarodriguez. (2021, July 21). La inteligencia artificial ayuda a estudiar la fauna de Doñana para mejorar su conservación - Fundación Descubre. Fundación Descubre. <https://fundaciondescubre.es/noticias/la-inteligencia-artificial-ayuda-a-estudiar-la-fauna-de-donana-para-mejorar-su-conservacion/>

- Daniel. (2023, October 30). U-NET : todo lo que tienes que saber sobre la red neuronal de Computer Vision. Formación En Ciencia De Datos | DataScientest.com. <https://datascientest.com/es/u-net-lo-que-tienes-que-saber>
- Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. (n.d.). Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. <https://arxiv.org/pdf/1506.01497>
- Saha, S. (2023, April 8). A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way. Medium. <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>
- SSD: Single Shot MultiBox Detector. (n.d.-a). Wei Liu¹ , Dragomir Anguelov² , Dumitru Erhan³ , Christian Szegedy³ , Scott Reed⁴ , Cheng-Yang Fu¹ , Alexander C. Berg¹. <https://arxiv.org/pdf/1512.02325>
- SSD: Single Shot MultiBox Detector. (n.d.-b). Wei Liu¹ , Dragomir Anguelov² , Dumitru Erhan³ , Christian Szegedy³ , Scott Reed⁴ , Cheng-Yang Fu¹ , Alexander C. Berg. <https://arxiv.org/pdf/1512.02325>
- Zheng, A. (n.d.). Evaluating Machine Learning Models. O'Reilly Online Learning. <https://www.oreilly.com/library/view/evaluating-machine-learning/9781492048756/ch03.html>

8. Anexos

8.1. Script Principal

En esta sección se presenta el script principal utilizado para implementar y ejecutar el modelo en el entorno de trabajo. Este script está diseñado para realizar diversas tareas clave, como la carga y preprocesamiento de los datos, la aplicación del modelo de detección o segmentación, y la evaluación de los resultados obtenidos.

El archivo del script principal se encuentra adjunto en formato .ipynb para que pueda ser revisado y ejecutado en un entorno Jupyter Notebook. Este archivo contiene el código completo necesario para ejecutar el flujo de trabajo, incluyendo la visualización de los resultados y la evaluación del desempeño del modelo.

El archivo adjunto es: script.ipynb.

8.2. Script unir_dataset

```
import os
import shutil

def consolidar_todas_imagenes(carpeta_base, carpeta_destino_todas, carpeta_destino_imagenes,
                             carpeta_destino_etiquetas):
    """
    Consolida todas las imágenes en una carpeta y organiza las etiquetadas en subcarpetas de
    `images` y `labels`.

    Args:
        carpeta_base (str): Ruta de la carpeta base que contiene subcarpetas con imágenes y
        etiquetas.
        carpeta_destino_todas (str): Carpeta donde se guardarán todas las imágenes.
        carpeta_destino_imagenes (str): Carpeta donde se consolidarán las imágenes etiquetadas.
        carpeta_destino_etiquetas (str): Carpeta donde se consolidarán las etiquetas
        correspondientes.
    """
    os.makedirs(carpeta_destino_todas, exist_ok=True)
    os.makedirs(carpeta_destino_imagenes, exist_ok=True)
    os.makedirs(carpeta_destino_etiquetas, exist_ok=True)

    contador_todas_imagenes = 0
    contador_imagenes = 0
    contador_etiquetas = 0
```

```

for carpeta_tarea in os.listdir(carpeta_base):
    ruta_tarea = os.path.join(carpeta_base, carpeta_tarea)
    if not os.path.isdir(ruta_tarea):
        continue

    directorio_imagenes = os.path.join(ruta_tarea, "images")
    directorio_etiquetas = os.path.join(ruta_tarea, "labels")

    if not os.path.exists(directorio_imagenes):
        print(f"Saltando carpeta: {carpeta_tarea} (falta 'images')")
        continue

    # Procesar imágenes
    for archivo_imagen in os.listdir(directorio_imagenes):
        ruta_imagen = os.path.join(directorio_imagenes, archivo_imagen)
        if os.path.isfile(ruta_imagen):
            # Copiar todas las imágenes a `carpeta_destino_todas`
            nuevo_nombre_imagen = f"{carpeta_tarea}_{archivo_imagen}"
            ruta_destino_todas = os.path.join(carpeta_destino_todas, nuevo_nombre_imagen)
            shutil.copy(ruta_imagen, ruta_destino_todas)
            contador_todas_imagenes += 1

            # Verificar existencia de etiqueta correspondiente
            if os.path.exists(directorio_etiquetas):
                archivo_etiqueta = archivo_imagen.replace('.JPG', '.txt').replace('.jpg',
'.txt')

                ruta_etiqueta = os.path.join(directorio_etiquetas, archivo_etiqueta)
                if os.path.isfile(ruta_etiqueta):
                    # Copiar imágenes etiquetadas a `images`
                    ruta_destino_imagen = os.path.join(carpeta_destino_imagenes,
nuevo_nombre_imagen)

                    shutil.copy(ruta_imagen, ruta_destino_imagen)
                    contador_imagenes += 1

                    # Copiar etiquetas correspondientes a `labels`
                    nuevo_nombre_etiqueta = f"{carpeta_tarea}_{archivo_etiqueta}"
                    ruta_destino_etiqueta = os.path.join(carpeta_destino_etiquetas,
nuevo_nombre_etiqueta)

                    shutil.copy(ruta_etiqueta, ruta_destino_etiqueta)
                    contador_etiquetas += 1

    print(f"Se consolidaron {contador_todas_imagenes} imágenes en: {carpeta_destino_todas}")
    print(f"Se consolidaron {contador_imagenes} imágenes etiquetadas en:
{carpeta_destino_imagenes}")

```

```
print(f"Se consolidaron {contador_etiquetas} etiquetas en: {carpeta_destino_etiquetas}")

if __name__ == "__main__":
    carpeta_base = r"C:\Users\crome\Desktop\TFM\Dades\osaconservation\Arribada2023GT"
    carpeta_destino_todas = r"C:\Users\crome\Desktop\TFM\Dades\osaconservation\Dataset_Completo"
    carpeta_destino_imagenes =
r"C:\Users\crome\Desktop\TFM\Dades\osaconservation\Dataset_Unificado\imagenes"
    carpeta_destino_etiquetas =
r"C:\Users\crome\Desktop\TFM\Dades\osaconservation\Dataset_Unificado\labels"
    consolidar_todas_imagenes(carpeta_base, carpeta_destino_todas, carpeta_destino_imagenes,
carpeta_destino_etiquetas)
```

8.3 Script Unir_Dividir_Manual

```
import os
import random
import shutil

def seleccionar_imagenes_aleatorias(carpeta_origen, carpeta_destino, porcentaje):
    """
    Selecciona un porcentaje de imágenes aleatorias de la carpeta origen y las copia a la carpeta
    destino.

    Args:
        carpeta_origen (str): Carpeta que contiene todas las imágenes.
        carpeta_destino (str): Carpeta donde se guardarán las imágenes seleccionadas.
        porcentaje (float): Porcentaje de imágenes a seleccionar (entre 0 y 100).
    """
    os.makedirs(carpeta_destino, exist_ok=True)

    # Listar todas las imágenes en la carpeta de origen
    todas_las_imagenes = [f for f in os.listdir(carpeta_origen) if f.lower().endswith(('.jpg',
'.jpeg', '.png'))]

    # Calcular la cantidad de imágenes a seleccionar
    cantidad_a_seleccionar = int(len(todas_las_imagenes) * (porcentaje / 100))

    # Seleccionar imágenes aleatoriamente
    seleccionadas = random.sample(todas_las_imagenes, cantidad_a_seleccionar)

    # Copiar las imágenes seleccionadas a la carpeta destino
    for imagen in seleccionadas:
        ruta_origen = os.path.join(carpeta_origen, imagen)
```

```

ruta_destino = os.path.join(carpeta_destino, imagen)
shutil.copy(ruta_origen, ruta_destino)

print(f"Se seleccionaron {cantidad_a_seleccionar} imágenes de {len(todas_las_imagenes)}
disponibles.")
print(f"Las imágenes seleccionadas se guardaron en: {carpeta_destino}")

#Función para seleccionar otro % de imagedes pero que no esten en la carpeta destino de la función
anterior, y se guarden en otra carpeta
def seleccionar_imagenes_aleatorias_2(carpeta_origen, carpeta_destino,
carpeta_comprobar_existencia, porcentaje):
    """
    Selecciona un porcentaje de imágenes aleatorias de la carpeta origen y las copia a la carpeta
    destino.

    Args:
        carpeta_origen (str): Carpeta que contiene todas las imágenes.
        carpeta_destino (str): Carpeta donde se guardarán las imágenes seleccionadas.
        carpeta_comprobar_existencia (str): Carpeta donde se comprobará si las imágenes
    seleccionadas ya existen.
        porcentaje (float): Porcentaje de imágenes a seleccionar (entre 0 y 100).
    """
    os.makedirs(carpeta_destino, exist_ok=True)

    # Listar todas las imágenes en la carpeta de origen
    todas_las_imagenes = [f for f in os.listdir(carpeta_origen) if f.lower().endswith(('.jpg',
'.jpeg', '.png'))]

    # Listar todas las imágenes en la carpeta de destino
    imagenes_destino = [f for f in os.listdir(carpeta_comprobar_existencia) if
f.lower().endswith(('.jpg', '.jpeg', '.png'))]

    # Calcular la cantidad de imágenes a seleccionar
    cantidad_a_seleccionar = int(len(todas_las_imagenes) * (porcentaje / 100))

    # Seleccionar imágenes aleatoriamente
    seleccionadas = random.sample(todas_las_imagenes, cantidad_a_seleccionar)

    # Comprobar si las imágenes seleccionadas ya existen en la carpeta destino y eliminarlas de la
    lista
    seleccionadas = [imagen for imagen in seleccionadas if imagen not in imagenes_destino]

    # Copiar las imágenes seleccionadas a la carpeta destino
    for imagen in seleccionadas:
        ruta_origen = os.path.join(carpeta_origen, imagen)

```

```

ruta_destino = os.path.join(carpeta_destino, imagen)
shutil.copy(ruta_origen, ruta_destino)

print(f"Se seleccionaron {len(seleccionadas)} imágenes de {len(todas_las_imagenes)}
disponibles.")
print(f"Las imágenes seleccionadas se guardaron en: {carpeta_destino}")

if __name__ == "__main__":
    # carpeta_origen = r"C:\Users\crome\Desktop\TFM\Dades\osaconservation\Dataset_Completo"
    # carpeta_destino = r"C:\Users\crome\Desktop\TFM\Dades\osaconservation\Para_Etiquetar_Huellas"
    # porcentaje = 20 # Cambia este valor para ajustar el porcentaje de imágenes a seleccionar
    # seleccionar_imagenes_aleatorias(carpeta_origen, carpeta_destino, porcentaje)

    carpeta_origen = r"C:\Users\crome\Desktop\TFM\Dades\osaconservation\Dataset_Completo"
    carpeta_destino = r"C:\Users\crome\Desktop\TFM\Dades\osaconservation\Para_Etiquetar_Huellas_2"
    carpeta_comprobar_existencia =
r"C:\Users\crome\Desktop\TFM\Dades\osaconservation\Para_Etiquetar_Huellas"
    porcentaje = 20 # Cambia este valor para ajustar el porcentaje de imágenes a seleccionar
    seleccionar_imagenes_aleatorias_2(carpeta_origen, carpeta_destino,
carpeta_comprobar_existencia, porcentaje)

```

8.4 Script Porcentajes:

```

import os
from datetime import datetime

def calcular_porcentaje_etiquetado(carpeta_imagenes):
    """
    Calcula el porcentaje de imágenes etiquetadas y cuántas quedan por etiquetar.

    Args:
        carpeta_imagenes (str): Ruta a la carpeta donde se encuentran las imágenes y JSONs.

    Returns:
        None
    """
    imagenes = [f for f in os.listdir(carpeta_imagenes) if f.lower().endswith(('.jpg', '.jpeg',
'.png'))]
    jsons = [f for f in os.listdir(carpeta_imagenes) if f.lower().endswith('.json')]

    total_imagenes = len(imagenes)
    total_etiquetadas = len(jsons)

```

```
# Calcular imágenes restantes
restantes = total_imagenes - total_etiquetadas
porcentaje = (total_etiquetadas / total_imagenes) * 100 if total_imagenes > 0 else 0

# Obtener el archivo JSON más reciente
if jsons:
    jsons_full_path = [os.path.join(carpeta_imagenes, f) for f in jsons]
    archivo_reciente = max(jsons_full_path, key=os.path.getmtime)
    nombre_reciente = os.path.basename(archivo_reciente)
    fecha_reciente = datetime.fromtimestamp(os.path.getmtime(archivo_reciente)).strftime('%Y-%m-%d %H:%M:%S')
else:
    nombre_reciente = "Ninguno"
    fecha_reciente = "N/A"

print(f"Total de imágenes: {total_imagenes}")
print(f"Imágenes etiquetadas: {total_etiquetadas}")
print(f"Faltan por etiquetar: {restantes}")
print(f"Porcentaje de etiquetado: {porcentaje:.2f}%")
print(f"Último archivo JSON modificado: {nombre_reciente} (Fecha: {fecha_reciente})")

if __name__ == "__main__":
    carpeta_imagenes = r"C:\Users\crome\Desktop\TFM\Dades\osaconservation\Para_Etiquetar_Huellas"
    calcular_porcentaje_etiquetado(carpeta_imagenes)
```