



Centro de Investigación Científica de Yucatán, A.C.

Posgrado en ENERGÍA RENOVABLE

AHORRO DE ENERGÍA DE UN ROBOT SUBMARINO IDENTIFICANDO AL PEZ LEÓN Y OPTIMIZANDO TRAYECTORIAS

Tesis que presenta

ÁNGEL TRINIDAD MARTÍNEZ GONZÁLEZ

En opción al título de

DOCTOR EN CIENCIAS EN ENERGÍA RENOVABLE

Mérida, Yucatán, México

Febrero 2022

POSGRADO EN CIENCIAS EN ENERGIA RENOVABLE



DECLARACIÓN DE PROPIEDAD

Declaro que la información contenida en la sección de Materiales y Métodos, los Resultados y Discusión de este documento proviene de las actividades de investigación realizadas durante el período que se me asignó para desarrollar mi trabajo de tesis, en las Unidades y Laboratorios del Centro de Investigación Científica de Yucatán, A.C., y que a razón de lo anterior y en contraprestación de los servicios educativos o de apoyo que me fueron brindados, dicha información, en términos de la Ley Federal del Derecho de Autor y la Ley de la Propiedad Industrial, le pertenece patrimonialmente a dicho Centro de Investigación. Por otra parte, en virtud de lo ya manifestado, reconozco que de igual manera los productos intelectuales o desarrollos tecnológicos que deriven o pudieran derivar de lo correspondiente a dicha información, le pertenecen patrimonialmente al Centro de Investigación Científica de Yucatán, A.C., y en el mismo tenor, reconozco que si derivaren de este trabajo productos intelectuales o desarrollos tecnológicos, en lo especial, estos se regirán en todo caso por lo dispuesto por la Ley Federal del Derecho de Autor y la Ley or la uso por la Ley Federal del Derecho de Autor y la Ley de la Propiedad industrial, en el tenor de lo expuesto en la presente Declaración.

Firma:

Ángel Trinidad Martínez González

Mérida, Yucatán, México, a 01 de Febrero de 2022

CENTRO DE INVESTIGACIÓN CIENTÍFICA DE YUCATÁN, A. C.

POSGRADO EN ENERGÍA RENOVABLE



RECONOCIMIENTO

Por medio de la presente, hago constar que el trabajo de tesis de Ángel Trinidad Martínez González titulado **"Ahorro de energía de un robot submarino identificando al pez león y optimizando trayectorias"**, fue realizado en la Unidad de Energía Renovable , Sistemas Híbridos de Energía, en el laboratorio de Sistemas Híbridos del Centro de Investigación Científica de Yucatán, A.C. bajo la dirección del Dr. Víctor Manuel Ramírez Rivera y codirección del Dr. David Antonio Gómez Jáuregui (Université de Bordeaux, Estla), perteneciente al Programa de Posgrado en Ciencias en Energía Renovable de este Centro.

Atentamente

Dra. Cecilia Hernández Zepeda

Directora de Docencia

Mérida, Yucatán, México, a 01 de Febrero de 2022

AGRADECIMIENTOS

Al Consejo Nacional de Ciencia y Tecnología (CONACYT) por la beca otorgada para la realización de esta investigación de doctorado, el complemento económico para la estancia académica realizada en Francia (Número de becario 444849) y por el financiamiento otorgado al proyecto 2015-01-786 para la solución de una problemática nacional.

Al Centro de Investigación Científica de Yucatán y al laboratorio de Sistemas Híbridos de Energía por el soporte de formación académica, por el uso de sus instalaciones y materiales proporcionados. Así como, al Instituto de Tecnología ESTIA (École Supérieure des Technologies Industrielles Avancées) en Francia, a su personal administrativo y colegas investigadores por su apoyo y hospitalidad durante mi estancia académica.

Al doctor Víctor M. Ramírez por la dirección de este trabajo, por el soporte en cada etapa del proceso, la comprensión y comunicación invaluables, además, del impulso y gestión para mi estancia académica en el extranjero. Al doctor David A. Gómez, por el seguimiento constante, aportaciones y críticas constructivas en los avances de la investigación, la gestión y acompañamiento en ESTIA, en el mismo sentido, por su guía y hospitalidad. A los doctores J. Adán Caballero y Adán Salazar por sus consejos, recomendaciones y aportaciones de gran valor para el desarrollo de esta tesis.

También a los técnicos Juan Méndez y Jorge Peniche por su apoyo en la disposición del equipo técnico y la toma de imágenes en el fondo marino. Así como, a Julia González M. por su apoyo técnico y administrativo.

A mi papá J. Trinidad por sus consejos acertados y el impulso que constantemente ha sembrado, al igual que mi mamá Angélica y mis Luis Alfredo, Guadalupe y Cecilia.

Un agradecimiento especial a mi esposa Jennifer Nhera por el apoyo y motivación incondicional para lograr este objetivo tan importante.

DEDICATORIA

Dedico esta investigación a mis hijos Yael y Naomi, quienes me impulsaron e inspiraron a continuar mi formación, quienes me han demostrado que todo se puede aprender y realizar.

ÍNDICE

List	a de Tabla	siv
List	a de Figura	iv
RE	SUMEN	vii
ABS	STRACT	viii
INT	RODUCCI	ÓN1
CAI	PÍTULO 1 .	
1.	PLANTEA	MIENTO DEL PROBLEMA
	1.1. AN	TECEDENTES
	1.1.1.	Abordaje del problema4
	1.2. JU	STIFICACION
	1.3. HIF	POTESIS6
	1.4. OB	JETIVOS6
	1.4.1.	General6
	1.4.2.	Específicos6
CAI	1.4.2. PITULO 2 .	Específicos
CAI 2.	1.4.2. PITULO 2 . ESTADO	Específicos
CAI 2.	1.4.2. PITULO 2 . ESTADO 2.1. SIS	Específicos
CAI 2.	1.4.2. PITULO 2 . ESTADO 2.1. SIS 2.1.1.	Específicos
CAI 2.	1.4.2. PITULO 2 . ESTADO 2.1. SIS 2.1.1. 2.1.2.	Específicos
CAI 2.	1.4.2. PITULO 2 . ESTADO 2.1. SIS 2.1.1. 2.1.2. 2.1.3.	Específicos
CAI 2.	1.4.2. PITULO 2 . ESTADO 2.1. SIS 2.1.1. 2.1.2. 2.1.3. 2.2. MC	Específicos
CAI 2.	1.4.2. PITULO 2 . ESTADO 2.1. SIS 2.1.1. 2.1.2. 2.1.3. 2.2. MC 2.2.1.	Específicos 6 7 7 DEL ARTE 7 STEMAS DE RECONOCIMIENTO DE PECES 7 Sistemas de DL en reconocimiento marino 8 Reconocimiento con condiciones no controladas 9 Sistemas de reconocimiento del pez león 11 INITOREO SUBMARINO 13 Métodos tradicionales de monitoreo 13
CAI 2.	1.4.2. PITULO 2 . ESTADO 2.1. SIS 2.1.1. 2.1.2. 2.1.3. 2.2. MC 2.2.1. 2.2.2.	Específicos .6 .7 .7 DEL ARTE .7 STEMAS DE RECONOCIMIENTO DE PECES .7 Sistemas de DL en reconocimiento marino .8 Reconocimiento con condiciones no controladas .9 Sistemas de reconocimiento del pez león .11 INITOREO SUBMARINO .13 Métodos tradicionales de monitoreo .13 Métodos de monitoreo con AUVs .15
CAI 2.	1.4.2. PITULO 2 . ESTADO 2.1. SIS 2.1.1. 2.1.2. 2.1.3. 2.2. MC 2.2.1. 2.2.2. 2.2.2	Específicos
CAI 2.	1.4.2. PITULO 2 . ESTADO 2.1. SIS 2.1.1. 2.1.2. 2.1.3. 2.2. MC 2.2.1. 2.2.2. 2.2.2 2.2.2	Específicos 6 7 7 DEL ARTE 7 STEMAS DE RECONOCIMIENTO DE PECES 7 Sistemas de DL en reconocimiento marino 8 Reconocimiento con condiciones no controladas 9 Sistemas de reconocimiento del pez león 11 ONITOREO SUBMARINO 13 Métodos tradicionales de monitoreo 13 Métodos de monitoreo con AUVs 15 1. Vehículos submarinos no tripulados 16 2. Procesos en el desplazamiento autónomo 17

	2.2.3. Vehículos submarinos aplicados al pez león	22
	2.2.4. Generación de caminos óptimos	24
	2.2.4.1. Los problemas de rutas en vehículos	24
	2.2.4.2. Problema del viajante de comercio (TSP)	25
	2.3. LIMITACIONES DEL ESTADO DEL ARTE	26
CA	\PITULO 32	27
3.	METODOLOGÍA DE LA GENERACIÓN AUTÓNOMA DE TRAYECTORIAS2	27
	3.1. EQUIPO UTILIZADO	29
	3.2. DISEÑO DEL SISTEMA DE GENERACIÓN AUTÓNOMA DE TRAYECTORIA	S 30
	3.2.1. Diseño del espacio marino en 3D	31
	3.2.2. Diseño del sistema de detección de estructuras marinas	34
	3.2.2.1. Visión estéreo del fondo simulado	35
	3.2.3. Localización de los puntos de búsqueda	37
	3.2.4. Transformación de coordenadas de pixeles a metros	
	3.2.5. Generación de ruta más corta	42
	3.2.5.1. Implementación y solución del TSP	43
	3.2.6. Método de evaluación del sistema generación autónoma de trayectorias	45
	3.2.6.1. Procedimiento para evaluar los modelos de líneas de intersección	45
	3.2.6.2. Procedimiento para evaluar los modelos propuestos con la	a ruta
	preestablecida	47
	3.3. RESULTADOS Y DISCUSIONES DEL CAPÍTULO	
	3.3.1. Resultados del sistema de generación autónoma de rutas óptimas	48
	3.3.2. Análisis y discusiones	53
	3.4. CONCLUSIONES DEL SISTEMA DE GENERACIÓN AUTÓNOMA	55
CA	\PITULO 45	8
4.	METODOLOGÍA DEL SISTEMA DE RECONOCIMIENTO DEL PEZ LEÓN5	8
	4.1. MODELOS PROBADOS PARA EL SISTEMA DE RECONOCIMIENTO	

4.1.1.	Descriptor HOG y clasificador SVM	58
4.1.2.	Redes Neuronales Convolucionales	59
4.1.3.	YOLOv2	60
4.2. IMF	PLEMENTACIÓN CON ALGORITMO YOLOv4 Y SSD-MobileNet-v2	61
4.2.1.	Equipo utilizado	62
4.2.2.	Base de datos	63
4.2.3.	Etiquetado de imágenes	64
4.2.4.	Implementación	64
4.2.5.	Procedimiento de validación cruzada	64
4.2.6.	Métricas para evaluar el desempeño	65
4.3. RE	SULTADOS Y DISCUSIONES DEL CAPÍTULO	66
4.3.1.	Resultados del sistema de reconocimiento del pez león	66
4.3.2.	Análisis y discusiones	71
4.4. CO	NCLUSIONES DEL SISTEMA DE RECONOCIMIENTO	73
CONCLUSION	IES GENERALES	74
TRABAJO FU	ruro	76
BIBLIOGRAFÍ	۹	77
ANEXOS		93

Lista de Tablas

Tabla 3-1. Resultados de distancias de rutas óptimas para los escenarios; separados por baja,
media y alta densidad51
Tabla 3-2. Distancias y porcentajes de diferencia entre las rutas de los escenarios de baja
densidad53
Tabla 3-3. Distancias y porcentajes de diferencia entre las rutas de los escenarios de densidad
media54
Tabla 3-4. Distancias y porcentajes de diferencia entre las rutas de los escenarios de densidad
alta55
Tabla 4-1. Resultados de validación cruzada de 5-veces en el entrenamiento con los sistemas
YOLOv4 y SSD-MobileNet-v267
Tabla 4-2. Resultados de las métricas de evaluación del conjunto y subconjunto de pruebas con
el modelo YOLOv467

Lista de Figuras

Figura 1-1. a) Pterois volitans y b) Pterois miles
Figura 1-2. Casos de mimetismo del pez león4
Figura 2-1. Ejemplo de foto-etiquetado de dos peces en el sistema de foto-identificación [54]11
Figura 2-2. Sistema de reconocimiento del pez león de Naddaf et al. (2018); a) interfaz de
usuario de la PC en la superficie y b) ejemplos de identificaciones correctas [55]12
Figura 2-3. Búsqueda exhaustiva o libre realizado por el equipo UCIA del CICY14
Figura 2-4. Método de monitoreo en zigzag siguiendo un transecto lineal [67]15
Figura 2-5 AUVs comerciales; a) Echo Ranger, b) Nessie IV y c) Remus 600017
Figura 2-6. AUVs militares; a) SeaOtter MKII, b) Bluefin-9 y c) Gavia Defence17
Figura 2-7. AUVs de investigación; a) Morpheus, b) Infante y c) SQX-50017
Figura 2-8. AUVs académicos y recreacionales; 2011 SONIA, a) Drekar y c) SubjuGator 717
Figura 2-9. Ejemplos de visualización de AUVs por medio de sonares; a) Sonar lateral y b)
sonar 3D de apertura sintética [82]18
Figura 2-10. Diagrama simplificado de la arquitectura de software de MEDUSA [84]20
Figura 2-11. Esquema de una trayectoria de una misión de MEDUSA [84]21
Figura 2-12. Ejemplos de uso de monitoreo en cuadrículas densas y anchas [89]21

Figura 2-13. Características del ROV (La Mar); a) diagrama de bloques de la sección en
superficie y submarina, b) imagen del ROV y descripción de partes [55]
Figura 2-14. Versión preliminar del vehículo submarino bio-inspirado [90]
Figura 2-15. Prototipo de ROV del Politécnico de Worcester [57]23
Figura 2-16. Vehículo ROV "The guardian LF1" [91]24
Figura 3-1. Momento de la generación de trayectorias en la metodología general27
Figura 3-2. Ejemplos para la trayectoria de censado del AUV; a) trayectoria para un vehículo
distinto al de monitoreo, b) trayectoria para el mismo vehículo que realiza el monitoreo29
Figura 3-3. Diagrama del sistema de generación autónoma de trayectorias
Figura 3-4. Elementos 3D de vegetación simulados en Blender
Figura 3-5. Elementos de rocas, arrecifes y corales simulados en Blender
Figura 3-6. Configuraciones de espacio simulado en Gazebo
Figura 3-7. Configuración gráfica de los nodos en el sistema ROS
Figura 3-8. Ejemplos de escenarios simulados en Gazebo
Figura 3-9. Configuración óptica de la visión estéreo
Figura 3-10. Imagen de profundidad a partir de 2 imágenes por visión estéreo37
Figura 3-11. Configuración de líneas de intersección
Figura 3-12. Apertura de los ángulos en una cámara40
Figura 3-13. Elementos que se relacionan en el arreglo óptico; a) corte horizontal y b) triángulo
generado hacia un punto de interés40
Figura 3-14. Transformación de coordenadas de pixeles a metros42
Figura 3-15. Obtención de rutas óptimas del escenario "10"; a) monitoreo completo y b)
monitoreo libre
Figura 3-16. Propuesta de sistema de visión para realizar monitoreo46
Figura 3-17. Trayectoria con distancia total de una ruta preestablecida supuesta para un AUV.
Figura 3-18. Sesgos generados por las líneas de intersección en escenarios de alta densidad
de elementos49
Figura 3-19. Sesgos generados por las líneas de intersección en escenarios de media densidad
de elementos
Figura 3-20. Sesgos generados por las líneas de intersección en escenarios de baja densidad
de elementos
Figura 3-21. Resultados del monitoreo libre (Mol-AUV) y minucioso (Mom-AUV) sobre el
escenario 6 de densidad baja de elementos52

Figura 3-22. Resultados del monitoreo libre (Mol-AUV) y minucioso (Mom-AUV) sobre el
escenario 23 de densidad media de elementos52
Figura 3-23. Resultados del monitoreo libre (Mol-AUV) y minucioso (Mom-AUV) sobre el
escenario 1 de densidad alta de elementos53
Figura 4-1. Momento en el que se realiza el reconocimiento del pez león en la metodología
general
Figura 4-2. Representación de los gradientes orientados; a) Cuerpo y espinas para el modelo 1,
b) Solo cuerpo para modelo 2 y c) Sólo cabeza para el modelo 359
Figura 4-3. Modelo implementado de Redes Neuronales Convolucionales (CNN) para el Pez
León60
Figura 4-4. Diagrama del sistema de detección del pez león con YOLOv260
Figura 4-5. Diagrama de los sistemas de detección de YOLOv4 y SSD-MobileNet-v2 con
Validación Cruzada de 5k62
Figura 4-6. Subconjuntos de prueba en relación a la iluminación presente en el pez; a)
Figura 4-6. Subconjuntos de prueba en relación a la iluminación presente en el pez; a) iluminación baja, b) media y c) alta63
Figura 4-6. Subconjuntos de prueba en relación a la iluminación presente en el pez; a) iluminación baja, b) media y c) alta
Figura 4-6. Subconjuntos de prueba en relación a la iluminación presente en el pez; a) iluminación baja, b) media y c) alta
Figura 4-6. Subconjuntos de prueba en relación a la iluminación presente en el pez; a) iluminación baja, b) media y c) alta
Figura 4-6. Subconjuntos de prueba en relación a la iluminación presente en el pez; a) iluminación baja, b) media y c) alta
Figura 4-6. Subconjuntos de prueba en relación a la iluminación presente en el pez; a) iluminación baja, b) media y c) alta
Figura 4-6. Subconjuntos de prueba en relación a la iluminación presente en el pez; a) iluminación baja, b) media y c) alta
Figura 4-6. Subconjuntos de prueba en relación a la iluminación presente en el pez; a) iluminación baja, b) media y c) alta
Figura 4-6. Subconjuntos de prueba en relación a la iluminación presente en el pez; a) iluminación baja, b) media y c) alta

RESUMEN

La llegada de especies a un hábitat del cual no son nativas, pueden provocar un deseguilibrio ecológico grave para el balance biológico de los ecosistemas. Pterois volitains y Pterois miles, ambas especies comúnmente llamadas "pez león" son considerados una plaga en las costas del Caribe en el mar Atlántico, provocando preocupación de gestores y científicos de la pesca y la conservación marina, por la posible afectación a la biodiversidad de las zonas involucradas. Los métodos tradicionales de observación y monitoreo marino son costosos, limitados y principalmente lentos. Para hacer frente a la problemática se ha propuesto el proyecto de diseño de un Vehículo Submarino Autónomo (AUV, por sus siglas en inglés Autonomous Underwater Vehicle) para monitoreo y localización del pez león; el cual contará con elementos que brindarán autonomía, como el banco de baterías y un sistema de visión artificial. La aportación de esta investigación es en el sistema de visión, la cual se divide en dos tareas principales: tarea 1; el reconocimiento del pez león mediante el sistema de reconocimiento basado en el algoritmo YOLOv4 (You Only Look Ones) y la tarea 2; la generación autónoma de trayectorias óptimas por medio de visión estereoscópica, técnicas de Procesamiento Digital de Imágenes (PDI) y TSP (Travel Salesman Problem) para encontrar el camino más corto. Estas tareas contribuirán a que el vehículo tenga un mayor grado de autonomía. El sistema de visión contará con cámaras de alta resolución y el dispositivo de procesamiento integrado Jetson Nano de NVIDIA.

En cuanto al sistema de reconocimiento del pez león, se han obtenido resultados de detección de 89.79% de precisión, 79.15% de mAP (mean Average Precision) y un 63.66% de sensibilidad con el sistema YOLOv4.

El sistema de generación de trayectorias óptimas fue probado en 42 escenarios mediante el simulador Gazebo, se diseñaron 2 modos de monitoreo; monitoreo libre (propuesta principal) y el otro minucioso o detallado, el primero obtiene un ahorro de distancia recorrida promedio del 52.49% menores que una ruta preestablecida, mientras el segundo modelo tiene un 21.56%, este ahorro de distancias se reflejará, por consecuencia, en un menor consumo de energía.

vii

ABSTRACT

The arrival of species to a habitat, of which they are not native, can cause a serious ecological imbalance for the biological balance of ecosystems. Pterois volitains and Pterois miles, both species commonly called "lionfish" are considered a pest on the Caribbean coasts in the Atlantic Sea, causing concern of managers and scientists of fisheries and marine conservation, due to the possible impact on the biodiversity of the areas involved. Traditional marine observation and monitoring methods are expensive, limited, and primarily time consuming. To face the problem, the project to design an Autonomous Underwater Vehicle (AUV) for monitoring and locating lionfish has been proposed; which will have elements that will provide autonomy, such as the battery bank and an artificial vision system. The contribution of this research is in the vision system, which is divided into two main tasks: task 1; recognition of the lionfish using the algorithm YOLOv4 (You Only Look Ones) recognition system and task 2; the autonomous generation of optimal trajectories through stereo vision, Digital Image Processing (PDI) and TSP (Travel Salesman Problem) techniques to find the shortest way. These tasks will contribute to the vehicle having a greater degree of autonomy. The vision system will feature high-resolution cameras and NVIDIA's Jetson Nano integrated processing device.

Regarding the lionfish recognition system, detection results of 89.79% precision, 79.15% mAP (mean Average Precision) and 63.66% sensitivity have been obtained with the YOLOv4 system.

The optimal trajectory generation system was tested in 42 scenarios using the Gazebo simulator, 2 monitoring modes were designed; free monitoring (main proposal) and the other meticulous or detailed, the first one obtains an average distance traveled savings of 52.49% less than a pre-established route, while the second model has 21.56%, this distance savings will be reflected, consequently, in lower power consumption.

viii

INTRODUCCIÓN

El pez león es el primer pez marino no nativo de la región, establecido a lo largo de la costa del atlántico de los Estados Unidos y del mar Caribe [1]. Los cambios en la abundancia y la composición de especies ocurren frecuentemente en las comunidades de peces que comparten áreas biogeográficas cercanas, esto como resultado de los movimientos migratorios relacionados con el clima y cambios oceanográficos. En consecuencia, existen afectaciones tanto para los conjuntos de organismos marinos como para la estructura y la dinámica de los mismos [2].

Es imperativo y necesario el estudio e investigación de este pez sobre las relaciones con el ecosistema, las afectaciones que podría ocasionar, el número de ellos y su localización. Actualmente, la cuantificación y monitoreo se hace por medio de dos técnicas de observación; la primera incluye censos visuales submarinos (UVC), video subacuático, teledetección, acústica y datos experimentales de captura y esfuerzo, la segunda técnica depende de la pesca, como datos de captura de pesquerías comerciales y recreativas [3]. Para cuantificar especies en grandes extensiones del arrecife en el mar Caribe, los procesos tradicionales se vuelven imprácticos, costosos, poco exactos y, por lo tanto, imposibles de realizar en su totalidad.

Una alternativa tecnológica para abordar esta problemática, es la propuesta en el proyecto 2015-01-786 de CONACYT, el cual propone el desarrollo y uso de un vehículo submarino autónomo con las propiedades tecnológicas necesarias para navegar en aguas profundas del Caribe mexicano, siguiendo una dinámica de exploración continua. Este AUV implementará, entre otras cosas, un sistema de visión artificial por medio de cámaras de alta definición para la detección del pez león y un sistema auto-generador de trayectorias óptimas para la navegación autónoma. El objetivo del uso de estos sistemas será optimizar el consumo de la reserva de energía del AUV y, a su vez, no depender del ser humano para diseñar trayectorias del vehículo y validar las detecciones del pez león, es decir, proveer una mayor autonomía.

La detección y reconocimiento de los peces se realiza a partir del uso de algoritmos o técnicas de PDI, el cual, es un tema de investigación desafiante en las ciencias computacionales, más aun, cuando se realiza en un lugar con condiciones de espacio e iluminación no controladas y con limitación de recursos computacionales [4]. Algunos de los métodos probados en este trabajo de investigación son: Histogram Oriented Gradients (HOG) con su clasificador Support Vector Machine (SVM) pertenecientes a los algoritmos tradicionales de aprendizaje automático

(ML, por Machine Learning); un sistema de reconocimiento del pez león adaptando como un clasificador a las Redes Neuronales Convolucionales (CNNs, por Convolutional Neural Networks) las cuales son la base del aprendizaje profundo (DL, por Deep Learning) para el reconocimiento de objetos; finalmente se comparan dos sistemas de reconocimiento de código abierto basados en DL como MovileNet-v2 y YOLOv4, entrenados e implementados en la tarjeta de desarrollo Jetson Nano de Nvidia.

La generación autónoma de rutas es implementada y diseñada a partir de 42 escenarios dentro de un entorno en tercera dimensión (3D) simulado en Gazebo sobre la plataforma del Sistema Operativo Robótico (ROS, por Robotic Operating System) y en la Jetson Nano, esto con la finalidad de que la transición a un entorno real sea con la misma configuración de comunicación, cámaras y en el mismo dispositivo hardware.

Los capítulos se distribuyen de la siguiente manera:

En el capítulo 1 se presenta el planteamiento del problema; información sobre los antecedentes del pez león, así como la justificación, hipótesis y objetivos de esta investigación. En el capítulo 2 se compone del estado del arte; son las principales técnicas y trabajos usados para resolver tareas similares. Para el capítulo 3 y 4 se describen las metodologías usadas para la generación de trayectorias autónomas y el sistema de reconocimiento, respectivamente, cada uno con sus respectivos resultados, análisis y conclusiones. Finalmente, se encuentran los apartados de Conclusiones generales, Trabajo Futuro, Bibliografía y Anexos.

CAPÍTULO 1

PLANTEAMIENTO DEL PROBLEMA

En este capítulo se presenta información relevante que define la problemática que pretende dar solución este trabajo de tesis. Los antecedentes nos brindan la información necesaria para conocer cómo inició y ha crecido la invasión del pez león, así como, lo realizado por la comunidad científica y voluntarios para mitigar la problemática. Luego, se presentan la justificación, hipótesis y objetivos de la investigación, los cuales nos permiten mostrar la forma en cómo podemos aportar una solución al problema.

1.1. ANTECEDENTES

Especímenes del pez león son ahora encontrados a lo largo de Este de los Estados Unidos, desde Cape Hatteras en Carolina del Norte hasta Florida, y en la Bermudas, las Bahamas, a través del Caribe, incluyendo Turks y Caicos, Haiti, Cuba, República Dominicana, Puerto Rico, San Croix, Belice y México [5]. Una de las teorías de cómo llegó el pez león al Atlántico es que este pudo ser liberado de un acuario accidentalmente en la playa de Biscayne en Florida, durante el Huracán Andrew en 1992 [6, 7].

Se ha encontrado que es un predador diurno, cuya mayor actividad es entre las 08:00 a las 11:00 hrs. Se alimenta, sobre todo, de peces en un volumen de 78% y de crustáceos en 14%. 21 familias y 41 especies de peces representan la dieta del pez león; las principales 10 familias de importancia son; Gobiidae, Labridae, Grammatidae, Apogonidae, Pomacentridae, Serranidae, Blenniidae, Atherinidae, Mullidae, y Monacanthidae [1]. Se han reportado varias poblaciones de pez león entre 0 a 60 metros de profundidad (arrecifes mesotróficos) y a mayor profundidad (150 metros), provocando la depredación, extinción o reducción de la biodiversidad local de sus principales presas [8].



Figura 1-1. a) Pterois volitans y b) Pterois miles

El pez león es un efectivo depredador de pequeños peces que viven en los arrecifes de coral, afectando por su actividad depredadora a la salud de estos últimos [9].



Figura 1-2. Casos de mimetismo del pez león.

El pez león posee un conjunto de rasgos que lo definen como un invasor, genera una interacción negativa con la fauna marina; tiene espinas venenosas para defenderse, una forma, un patrón de coloración (Figura 1-1) y comportamiento que lo ayudan a tener un fuerte mimetismo con el lecho marino (Figura 1-2), alta habilidad competitiva, baja vulnerabilidad a parásitos, presenta un rápido crecimiento en su tamaño y una taza elevada de reproducción. El peor escenario, si no se hace nada al respecto, son los efectos directos e indirectos del pez león combinados con los impactos preexistentes de estrés (especialmente la sobrepesca), causaría un substancial e irreparable deterioro en las comunidades de arrecifes de coral [10].

1.1.1. Abordaje del problema

Al identificarse el pez león por primera vez en los años 90s en Florida, inició una alerta hacia el sector pesquero, instituciones académicas y a organizaciones civiles para identificar y realizar búsquedas de la especie en la zona y sus alrededores [11, 12]. Desde entonces ha sido constante el monitoreo, el análisis de su expansión y el daño que puede causar la especie invasora. En consecuencia, en el 2010 se celebró el Taller Regional sobre el pez león en Cancún (México) en el que se definieron un conjunto de prioridades en materia de investigación, entre las que figuraban las siguientes [13]:

- i) Cuantificar las repercusiones socioeconómicas de la invasión del pez león,
- ii) Crear un modelo de la expansión futura de la invasión, que incluya la dispersión de las larvas,
- iii) Realizar investigaciones aplicadas sobre la captura incidental del pez león,

- iv) Realizar investigaciones aplicadas para estudiar eficazmente al pez león en aguas profundas,
- v) Normalizar los métodos de investigación,
- vi) Determinar la eficacia de las medidas de extracción,
- vii) Hacer hincapié en el peligro o en la ausencia de peligro de la ciguatera en el pez león,
- viii) Examinar los patrones de conectividad de la población del pez león,
- ix) Realizar monitoreo espacial del pez león (en términos de espacio),
- x) Investigar la esterilización como método de control, y
- xi) Estudiar la interacción entre las especies nativas y las invasoras.

Este trabajo de investigación pretende contribuir con las medidas *iv*, *v* y *ix* propuestas en el Taller Regional.

1.2. JUSTIFICACION

El Caribe mexicano, considerado uno de los ecosistemas más importantes para la conservación de los peces y la biodiversidad en el mundo, se encuentra amenazado en su equilibrio ecológico debido a la invasión del pez león que actualmente se reproduce libremente debido a la ausencia de depredadores, por lo que se requiere hacer un estudio de cuantificación, monitoreo y localización de la especie, para cuantificar el efecto de la invasión sobre todo en profundidades mayores a 30 metros y en zonas de difícil acceso.

Una de las principales dificultades para hacer un análisis cuantitativo del problema de la invasión del pez león, es la localización de poblaciones, ya que se encuentran en zonas de gran profundidad y de difícil acceso. Los métodos convencionales (inmersión por buceo) aportan una serie de datos limitados como consecuencia de las restricciones físicas del ser humano para descender a profundidades mayores de 20 metros, las horas que pueden bucear y el área que cubrir, considerando que el Caribe mexicano cuenta con 5.7 millones de hectáreas marítimas (aunque posiblemente se tomen muestras del territorio), son incapaces de generar mapas de distribución a gran escala, precisos y representativos. Para llevar a cabo esta tarea es excesivamente caro; con los costos de los buzos, los barcos o lanchas, el obstáculo de las cuestiones meteorológicas, además, de la seguridad y salud de los buzos. Por lo tanto, tiene sentido diseñar un método alternativo, rentable y seguro para realizar estas tareas.

Aunque los trabajos recientes han alcanzado resultados satisfactorios para especies variadas, aún no existen sistemas de reconocimiento para la detección confiable del pez león. Se pueden probar arquitecturas de reconocimiento de objetos existentes que pueden detectar múltiples objetos en tiempo real.

Por otro lado, para que el vehículo no dependa de un control humano o de rutas preestablecidas, debe tener la capacidad de tomar decisiones sobre la ruta que debe seguir en busca de peces león, debido a que la búsqueda será por un sistema de visión artificial simulando un comportamiento humano. Estos lugares ayudarán a que el vehículo sea más eficiente en su desplazamiento y, en consecuencia, tendrá un menor gasto de energía.

1.3. HIPOTESIS

1. Implementando métodos de visión artificial adecuados se podrá detectar, en tiempo real, el pez león de forma precisa en condiciones no controladas.

2. Con la generación autónoma de trayectorias óptimas se minimizará el consumo energético de las fuentes de energía renovable con las que se alimenta el vehículo hasta en un 50% durante el monitoreo, en comparación con una ruta pre-establecida.

1.4. OBJETIVOS

1.4.1. General

Identificar el pez león mediante algoritmos de visión artificial y generar trayectorias óptimas de navegación con el propósito de minimizar el consumo de energía de los sistemas renovables y de almacenamiento del vehículo submarino.

1.4.2. Específicos

- Desarrollar un sistema de visión por computadora utilizando arquitecturas y algoritmos de reconocimiento de objetos que más se adapten para el reconocimiento del pez león en condiciones no controladas.
- Desarrollar un sistema de detección de formas o estructuras del fondo marino a partir de visión estéreo y PDI.
- Diseñar un método para encontrar coordenadas geográficas o espaciales (en metros) de los puntos de interés del fondo marino a partir de una imagen.
- Generar trayectorias óptimas para los puntos de interés con algoritmos voraces.

CAPITULO 2

ESTADO DEL ARTE

La visión por computadora hace frente al problema de la ingeniería de sistemas de visión artificial, capaz de comprender e interpretar nuestra realidad. Los temas populares en este campo incluyen reconocimiento de escena, reconocimiento de objetos, interpretación de movimiento, navegación autónoma y manipulación robótica de objetos en una escena [14]. Este capítulo muestra el estado de la técnica divididos en 2 apartados en relación a los dos problemas a resolver en este trabajo de investigación; la primera parte está dedicada los sistemas de reconocimiento aplicados a peces, desde los algoritmos tradicionales, hasta los basados en CNN llamados también DL, además, se muestran algunos sistemas aplicados al reconocimiento del pez león. En la segunda parte, se presentan trabajos relacionados al monitoreo marino usando técnicas tradicionales de buceo y técnicas modernas mediante AUV, también, se muestran vehículos operados remotamente (ROV, por Remote Operated Vehicle) diseñados o adaptados para monitoreo del pez león, finalmente, algunos trabajos relacionados con los problemas de ruta de vehículos (VRP, por Vehicle Route Problem).

2.1. SISTEMAS DE RECONOCIMIENTO DE PECES

El problema de un sistema de reconocimiento puede ser abordado desde diferentes puntos de vista para su solución. Se debe tener en claro el objeto a detectar, el espacio donde se encuentra, puntos de interés en los objetos, distancias y ángulos. El reto más grande es relacionarlo con una categoría o clase de reconocimiento, debido a que existen objetos muy variados en su forma como puede ser un perro y un mueble. En muchos casos, el reconocimiento depende del contexto alrededor de los objetos y los elementos de la escena [15].

La investigación en el campo de la visión por computadora está minuciosamente estudiada y avanzada. Muchos de los algoritmos hechos para la detección de objetos se están utilizando en el entorno submarino, desde técnicas simples como la medición del tamaño de los peces a través de distancias con puntos estratégicos [16], hasta aplicaciones más complejas como la detección de peces y el monitoreo de escenas. ([17]; [18]; [19]) con la ventaja del uso de cámaras fijas en el fondo del mar.

Varias especies de peces están entre las más importantes tareas de observación oceánica, beneficiando el trabajo de investigadores académicos como biólogos y científicos oceánicos [20].

Los estudios en peces son ampliamente aplicados a la morfología y calidad [21, 22], comportamiento [23, 24], detección de enfermedades [25, 26], así como aplicaciones de clasificación y detección de peces con algoritmos tradicionales de ML como Hidden Markov Models (HMMs) [27], análisis de componentes principales (PCA, por Principal Components Analysis) [28], modelo de textura gaussiana (GMM, por Gaussian Mixture Model) [29, 30], histograma de gradientes orientados (HOG, por Histogram Oriented Gradients) [28, 31].

Por mencionar algunos ejemplos específicos; un trabajo de Spampinato [27] y colaboradores, al igual que Boom [32] presentan un sistema para analizar imágenes de cámaras fijas subacuáticas a largo plazo (más de 3 años de 12 horas al día de tomas submarinas de 10 cámaras). Este sistema contribuirá en el estudio de los efectos que el cambio climático y la contaminación tienen en el medio ambiente conformando un proyecto llamado "Ecogrid proyect", mediante la clasificación de peces y el seguimiento de trayectorias (comportamiento); combinan características de textura como estadística del histograma en niveles de gris, filtrado espacial de Gabor y propiedades de la matriz de co-ocurrencia, además de usar características de forma como la transformada del espacio-escala de curvatura y el histograma de descriptores de frontera de Fourier; los resultados se obtuvieron con una base de datos de 360 imágenes de 10 especies de pez, obteniendo una precisión del 92%, además de registrar algunos comportamientos de su trayectoria. Continuando con los objetivos de Ecogrid proyect, ahora llamado "Fish4Knowledge1", Hsiao, Huang y Boom en 2014, usaron el mismo sistema de reconocimiento, pero con diferentes clasificadores y sin seguimiento de trayectoria; los resultados de eficiencia para el primer autor con clasificación de representación dispersa (SCR-MP) es entre 81.8% y 96%, para el segundo con clasificador BGOT es de 86.75% y, para el tercer autor con un clasificador SVM fue de 76.71% [18, 31, 29].

2.1.1. Sistemas de DL en reconocimiento marino

Como resultado de lograr mayor eficiencia en los procesos, técnicas y algoritmos de ML, surgieron los algoritmos de aprendizaje profundo o DL (por Deep Learning) cuya base son las Redes Neuronales Artificiales (ANN, Artificial Neural Networks) [33, 34], también son utilizados para la clasificación de imágenes de algunas especies de peces. Otros ejemplos específicos, en industrias acuícolas usaron un modelo de DL fusionado con Naive Bayesian para el

reconocimiento de peces en un ambiente controlado, teniendo resultados hasta del 98.94 para seis diferentes especies [35]. Qin y colaboradores en 2016 [30], usaron un modelo CNN con SVM para el reconocimiento de peces en un ambiente controlado, teniendo resultados de eficiencia del 98.64%.

Además, surgieron arquitecturas más complejas de DL como AlexNet [36], GoogLeNet (Inception) [37], ResNet [38], CNN regional (R-CNN) [39], Fast R-CNN [40], R-CNN más rápido [41]; los tres últimos basados en AlexNet y de alta velocidad a la hora de analizar imágenes mediante Unidad de Procesamiento Gráfico (GPU), cuyos usos se están expandiendo actualmente a todo tipo de aplicaciones y entornos. En 2016 [42] utilizaron y compararon los métodos de Inception de GoogLeNet's con el método tradicional HOG-SVM para reconocer peces de arrecifes, demostrando la enorme ventaja de los métodos de DL del 43% sobre el método tradicional. En otra investigación [43], también se utilizó un modelo CNN de GoogLeNet's para reconocer 20 especies de peces y comparar con la capacidad del ser humano para reconocer los peces, teniendo un 94.9% ganando a los humanos que obtuvieron un 89.3%. En otro trabajo de investigación [44] aplicaron el modelo AlexNet para el reconocimiento de peces, logrando una precisión del 99,43%.

La mayoría de los trabajos anteriores usando las arquitecturas ML, tanto las tradicionales, así como, las más recientes basadas en DL, se diferencian considerablemente en su desempeño en condiciones similares de ambiente controlado, siendo las recientes arquitecturas las mejor desempeñadas, como se observa en el trabajo [42] que compara directamente los métodos y, en otro ejemplo claro, en [41] usando arquitectura DL que supera a los trabajos hechos anteriormente con métodos tradicionales en la misma base de datos dentro del proyecto "Fish4Knowledge1". Inclusive, las arquitecturas DL abren la puerta a desafíos como el reconocimiento en condiciones no controladas.

2.1.2. Reconocimiento con condiciones no controladas

Es un gran reto identificar al pez león, en general, a cualquier especie o elemento marino, especialmente cuando el entorno no está controlado. Esto significa que la iluminación, el fondo, la orientación de los objetos, las oclusiones y el movimiento de la cámara no son modificados para facilitar la identificación [45]. El objetivo de controlar el entorno significa separar el objeto del fondo, por lo que es posible estimar la región de interés más probable en la imagen. Después de eso, será posible aplicar técnicas de reconocimiento o clasificación para lograr

menos errores o identificaciones falsas, especialmente cuando el fondo es tan variado con el lecho marino [46, 47].

Los estudios en los que se fija la cámara permiten el uso de una técnica llamada sustracción de fondo [48] así como técnicas como la observación de cambios en los niveles de intensidad de píxeles o cambios en la textura (como GMM) entre la nueva imagen y una anterior o imagen de referencia. Estas técnicas se utilizan en la detección de peces en [49], lo que permite distinguir los objetos en movimiento del entorno estático. Además de eso, en [44] (mencionado anteriormente) donde usaron la arquitectura AlexNet para el reconocimiento de peces, con la ayuda de la cámara fija y la técnica de extracción de primer plano lograron la precisión del 99,43%. Estos ejemplos mostrados, lograron adaptar un espacio no controlado para que sus arquitecturas tuvieran mejor desempeño.

Algunos proyectos aún están en proceso de aplicar un sistema de reconocimiento en un entorno incontrolado y complicado como el fondo marino, además de tener una cámara en movimiento [50, 51, 52]. Por ejemplo, en 2016 Chuang y colaboradores en [50], utilizan un entorno natural con la ayuda de un robot en movimiento, por lo que tener el control del fondo como en los ejemplos anteriores es imposible. Sin embargo, realizan la identificación de forma vertical y mediante una potente iluminación que le da un alto contraste al pez de enfrente y cerca de él, provocando un fondo oscuro, suave y casi imperceptible. Siddiqui y su equipo en 2018, aún se encuentran en un sistema de pre-implementación para la identificación con múltiples detecciones de peces; trabajan con imágenes de peces en su entorno natural, no buscan una región de interés automáticamente, sino que, eligen la coordenada del pez manualmente y, a partir de ahí, se busca dar seguimiento con el reconocimiento [51]. Los autores en [52] desarrollaron un sistema de identificación de peces con sonar acústico basado en un entorno simulado. La distancia entre el fondo del mar y los peces permite al sonar obtener imágenes contrastantes entre ambos elementos, facilitando su separación.

La aparición de las CNN ha revolucionado por completo la forma en que se realizan las tareas de reconocimiento de imágenes, pues ya no es necesario controlar el entorno para conseguir un mejor rendimiento. Modelos como Inception, ResNet, R-CNN, Fast R-CNN, Faster R-CNN, SSD-MobileNet, You Only Look Ones (YOLO) [53] están integrados con herramientas de software como Tensorflow, Pytorch, Café, TensorRT y algunos con dispositivos de hardware GPU. Estos, software y hardware, ayudan a organizar y ejecutar programas en paralelo, lo que reduce considerablemente el tiempo de procesamiento. A diferencia de los algoritmos ML tradicionales, se ha demostrado que los algoritmos DL funcionan mejor en entornos complejos

donde la iluminación, la orientación de los objetos, las distancias y el fondo no son controlables. En este sentido, los algoritmos DL pueden resolver el desafío de reconocer un pez con un alto grado de mimetismo en un entorno marino complejo.

2.1.3. Sistemas de reconocimiento del pez león

En esta sección, destacamos cuatro estudios que abordan la implementación de sistemas de visión por computadora aplicados al pez león; cada uno presenta un objetivo o enfoque diferente.

En 2016 se desarrolló un sistema de foto-identificación del pez león para su identificación individual en tareas de seguimiento [54], con el fin de realizar estudios de comportamiento en trabajos posteriores. Se utiliza una técnica llamada foto-etiquetado para comparar patrones dentro de un área delimitada por puntos marcados manualmente en imágenes del pez león (Figura 2-1), es decir, cada vez que se quiere identificar un pez, se debe tomar una imagen de costado, usar la interfaz de usuario para etiquetar con puntos las áreas del pez, luego el sistema puede identificar el pez, si ha sido registrado en la base de datos. Este método está dirigido a científicos que estudian individuos de la especie para identificarlos a través del tiempo. No se encontró una continuación de este trabajo.





En 2018 Naddaf y colaboradores utilizaron un vehículo operado por control remoto (ROV) para ayudar a los buzos que cazan peces león con lanza [55]. Una cámara instalada en una tarjeta arduino envía imágenes a la superficie donde una computadora con una GPU NVIDIA GTX 745 las procesa usando el modelo de detección R-CNN. Se presenta una detección positiva del 92% para un video que contiene 500 imágenes con una confianza de detección mínima del 85% de resultados. Sin embargo, carece de un detallado análisis para evaluar la capacidad real de

su sistema de detección del pez león, se pueden observar ejemplos de detección en la Figura 2-2. En los capítulos siguientes se detalla el procedimiento y las métricas que se evalúan en un sistema de reconocimiento de objetos.



Figura 2-2. Sistema de reconocimiento del pez león de Naddaf et al. (2018); a) interfaz de usuario de la PC en la superficie y b) ejemplos de identificaciones correctas [55].

Los autores Furlan y Boniface en 2018, en el marco de un proyecto de la universidad Politécnica de California para combatir la invasión del pez león, utilizaron el marco de Google *Inception* como clasificador de imágenes para siete especies de peces, incluido el pez león [56]. Informan tener resultados 100% positivos en la prueba para la clasificación de 100 imágenes. Sin embargo, al tratarse de un sistema clasificador entre una especie y otra, tener pocas imágenes para realizar una evaluación y no mostrar el proceso detallado de la evaluación, se considera un sistema sobre-ajustado y no se puede tener comparación con otros trabajos.

En el reporte final [57] visto con anterioridad, los autores informaron en su sistema de visión la detección de pez león con tres sistemas de reconocimiento: Inception v2, SSDlite-Mobilenet-v1 y SSDlite-mobilenet-v2, con una media del promedio de precisión (mAP) del 24%, 20% y 31,5% respectivamente. No se ha encontrado una continuación de este trabajo por parte de los autores, en la que reporten una mejora en sus resultados.

2.2. MONITOREO SUBMARINO

2.2.1. Métodos tradicionales de monitoreo

La forma de enfrentar esta problemática ha variado conforme evoluciona la situación de la invasión y la magnitud del daño. Lo que no ha cambiado mucho son los actores que participan; ciencia ciudadana es un término que describe la colaboración entre ciudadanos voluntarios y científicos, generalmente apoyando la conservación con datos y mano de obra para el monitoreo y evaluación [58]. En los inicios de la invasión para el proceso de monitoreo del pez león, los pescadores participaron activamente reportando avistamientos durante sus actividades de pesca [59]. Debido a las características físicas del pez león, no puede detectarse con tecnologías de detección masiva, como los sonares. Los métodos de monitoreo aplicables al pez león necesariamente deben ser visuales, debido a sus características físicas, es decir, que sea identificado de manera visual y directa por el ojo humano (forma tradicional o convencional) o por sistemas de procesamiento de imágenes con reconocimiento de objetos (forma tecnológica actual); en este apartado escribiremos algunos métodos utilizados de manera convencional llamados estudios de censos visuales subacuáticos (UVC por Underwater Visual Cencus) [60] para la búsqueda y monitoreo de especies marinas, también usados para el monitoreo del pez león, puede realizarse con esnórquel o en buceo SCUBA (Self Contained Underwater Breathing apparatus).

Un método comúnmente usado en estudios UVC, no necesariamente en monitoreo formal, es la *búsqueda exhaustiva*; un método usado por un equipo de investigadores y voluntarios para el monitoreo del pez león en la reserva marina de Bacalar Chico, Belice [61] y por nuestro equipo de la Unidad de Ciencias del Agua (UCIA) en Cancún (Figura 2-3). Es una búsqueda sin rutas establecidas, es decir, una búsqueda libre. Este tipo de monitoreo permite una búsqueda más detallada de la especie, sobre todo, en algunos micro hábitats complicados como en arrecifes, cuevas y grietas.



Figura 2-3. Búsqueda exhaustiva o libre realizado por el equipo UCIA del CICY.

Los transectos lineales es el método comúnmente usado en estudios UVC para la búsqueda de especies marinas [62]. Algunos estudios para monitorear al pez león hacen uso de este método; consiste en marcar líneas rectas paralelas entre sí, a una distancia de entre 1 a 3 metros, los buzos realizan un recorrido sobre ella, buscando a ambos lados de entre 0.5 a 2 metros [63, 64, 65]. La anchura de un transecto es generalmente entre 1 y 5 metros, depende de la complejidad del hábitat y en la confianza que se tiene para avistar las especies [66]. Funciona adecuadamente el método en micro hábitats no complejos como en sedimentos suaves, arenas y pastos pequeños.

Otro equipo de investigadores usó un método UVC de *transecto lineal* en Florida [67], le llamaron búsqueda centrada del pez león (LFS por Lionfish Focused Search); la aplicación del método consistió en realizar un recorrido en "zigzag" siguiendo un transecto lineal, previamente marcado con una cinta, en tramos de 25 metros y un ancho de 10 metros (Figura 2-4). Otro método adaptado para búsqueda del pez león fue hecho por el equipo anteriormente mencionado de Belice [61], donde tomaron el método LFS y lo combinaron con el método de búsqueda libre; consistió en seguir el transecto lineal de 30 metros y un ancho de 5 metros, con un patrón en "U", realizando una búsqueda minuciosa en el área.



Figura 2-4. Método de monitoreo en zigzag siguiendo un transecto lineal [67]

2.2.2. Métodos de monitoreo con AUVs

Las cámaras de video se utilizan para obtener imágenes del lugar donde los ROVs están realizando la operación o en los AUVs para tomar fotografías o videos de la zona que se está explorando. Otra aplicación de los sistemas de visión es obtener estimados del movimiento relativo y en algunas ocasiones absoluto usando un algoritmo tipo SLAM, esto es útil para realizar tareas como lo son el seguimiento por visión de tuberías y el control de la posición [68]. La tecnología de los sistemas de visión óptica se encuentra muy desarrollada, esto permite la obtención de imágenes con alta resolución, colores definidos y el procesamiento en tres dimensiones. Sin embargo, a partir de cierta distancia, que dependerá del grado de turbidez del agua, los objetos submarinos son difícilmente distinguibles. Es por esto que la mayoría de investigaciones se han enfocado en la búsqueda de alternativas para producir imágenes submarinas mediante el uso de ondas acústicas, debido a que el sonido presenta menor atenuación que la luz en el agua [69].

Ejemplos de desarrollos de estos tipos de robots son los proyectos ALIVE y RAUVI [70 y 71]. El vehículo ALIVE navega autónomamente hasta el lugar donde se realizará la intervención, una vez que llega a la posición deseada, cambia su operación a Control Supervisado y, mediante comunicación acústica, realiza las tareas de manipulación. Por su parte, el robot GIRONA 500 (del proyecto RAUVI) explora primero la región de interés y toma información acústica y visual del fondo. Posteriormente el robot sube a la superficie donde la información recolectada es procesada para realizar una reconstrucción de la región explorada. A través de una interfaz gráfica de usuario un operador identifica el objeto de interés y da los comandos al robot para que realice la intervención. Hasta el momento se han realizado exitosamente pruebas

experimentales en un escenario en el que la misión del robot era recuperar una caja negra de avión [72].

2.2.2.1. Vehículos submarinos no tripulados

Más del 70% de la superficie del planeta está bajo el agua, adicionalmente, la mayor parte del fondo marino es todavía un territorio desconocido [73]. Desde hace décadas, cuando aparecieron los primeros vehículos submarinos no tripulados (UUV por Unmanned Underwater Vehicle), se deseaba tener alternativas más confiables, seguras y eficaces para realizar las tareas; intentando cambiar y colaborar con las formas tradicionales de observar, estudiar e investigar acerca del océano [74]. Los ROVs han sido, desde la aparición de los primeros UUV, los más utilizados para enfrentar los desafíos que se presentan en las tareas submarinas; son manipulados a distancia por un usuario, esto significa, el operador toma todas las decisiones desplazamiento, la ejecución de acciones del vehículo y el envío de datos [75]. En cambio, los AUVs son robots que operan bajo el agua y no requieren intervención de un operador humano para la toma de decisiones en cuanto a su desplazamiento en operaciones de misión, en su lugar, el vehículo extrae información de su entorno empleando una amplia variedad de sensores, y luego emplea dicha información para desplazarse en el medio marino [76].

En este sentido, los AUVs tienen el potencial de revolucionar nuestro acceso a los océanos, abordando problemas críticos a los que se enfrenta la comunidad científica dedicada a la oceanografía y al campo marino, tales como búsqueda y cartografía de los lechos marinos, evaluación del cambio climático y vigilancia del hábitat marino entre otras [73].

Específicamente las aplicaciones de los AUVs, hasta hace poco, habían sido limitadas por la tecnología disponible. Con el desarrollo de mayores capacidades de procesamiento y baterías de mayor capacidad, los AUVs actualmente son empleados en más tareas, con roles y misiones que evolucionan constantemente [77].

Existen desarrollos de vehículos de forma comercial que pueden realizar trabajos de estudios del lecho marino, aplicaciones de revisión e instalación de tuberías submarinas y reparación de infraestructuras (Figura 2-5). En aplicaciones de tipo militar para localizar minas submarinas, patrullar una zona y en la detección de submarinos (Figura 2-6). En investigación para realizar estudios de playa, vigilancia y evaluación del medio ambiente y estudios de ciencia marina (Figura 2-7). Finalmente, en la Figura 2-8 se observan los de tipo académico y recreacional, creados por estudiantes para competir en entornos generalmente cerrados [78].



Figura 2-5 AUVs comerciales; a) Echo Ranger, b) Nessie IV y c) Remus 6000.



Figura 2-6. AUVs militares; a) SeaOtter MKII, b) Bluefin-9 y c) Gavia Defence.



Figura 2-7. AUVs de investigación; a) Morpheus, b) Infante y c) SQX-500.



Figura 2-8. AUVs académicos y recreacionales; 2011 SONIA, a) Drekar y c) SubjuGator 7.

2.2.2.2. Procesos en el desplazamiento autónomo

Los robots submarinos están equipados con sistemas de sensores dedicados a determinar la localización del vehículo, su velocidad y aceleración, así como su estado de funcionamiento. Los sistemas de visión entran en esta categoría de dispositivos cuando son utilizados para determinar la posición y orientación de objetos con respecto al vehículo. Los sensores permiten controlar los movimientos del robot lo cual a su vez permite que este pueda realizar la misión que le fue encomendada. Huvenne y su equipo en 2018, clasifican los AUV como vehículos "de crucero" o "flotantes" [78]; los AUV de crucero generalmente tienen forma de torpedo, impulsados por una sola hélice, optimizados para cubrir grandes distancias pero carecen de

habilidad vertical, en cambio los flotantes están equipados con varias hélices, que facilitan la capacidad de maniobrabilidad multidireccional. Los AUV flotantes están diseñados para operaciones de precisión, levantamientos en cámara lenta (por ejemplo, fotografía de los fondos marinos) y funcionan en terrenos claramente tridimensionales, como alrededor de arrecifes de alto relieve [79].

A continuación, se presentan algunos sensores que utilizan la mayoría de los vehículos submarinos.

GPS. Permite determinar la posición de un objeto en cualquier parte del mundo. Estos se utilizan en las embarcaciones que apoyan en la misión de un ROV o en los Planeadores acuáticos para localizarse una vez que emergen a la superficie del mar. No puede ser usado por debajo del agua [80].

Velocímetro Doppler de navegación. Se basa en la transmisión de tres o cuatro ondas ultrasónicas, con dirección diagonal hacia el fondo, ligeramente inclinado a los lados, al frente y hacia atrás. El dispositivo cuenta con un circuito de procesamiento que, mediante el análisis de los ecos recibidos y sus correlaciones, calcula la velocidad, y la posición se obtiene indirectamente por odometría. La operación del velocímetro de Doppler está limitada a un cierto nivel con respecto al fondo marino [81].

Sonar para detección de obstáculos. Está basado en varios proyectores y arreglos de hidrófonos (Figura 2-9). En función de la diferencia de tiempo entre la emisión de la señal acústica y el eco recibido se determina la distancia a la que se encuentra un objeto al frente o en ciertos ángulos. Para la detección de obstáculos también es posible utilizar emisores láser y sus receptores, pero estos pueden presentar problemas en aguas muy turbias [82].



Figura 2-9. Ejemplos de visualización de AUVs por medio de sonares; a) Sonar lateral y b) sonar 3D de apertura sintética [82].

2.2.2.3. Control de trayectorias en AUVs

En el trabajo de Antonelli se presenta el concepto del Sistema de Control de la Misión (MCS por Mision Control System); es el conjunto de programas responsables de que se lleve a cabo la misión deseada, está usualmente equipado con una interfaz gráfica que permite al operador comandar diferentes tareas que componen la misión, contiene los elementos de Guiado, Control y Navegación del robot [83]. Se muestra un ejemplo de este sistema con el AUV MEDUSA [84] en la Figura 2-10. Las tareas de la misión son generalmente concurrentes y su manejo depende del estado del vehículo y de las condiciones ambientales; por lo tanto el MCS maneja las tareas, eventualmente suprimiéndolas y definiendo su secuencia (modificando y priorizando). El MCS está compuesto por los siguientes sistemas:

- Sistema de Guiado. Este genera las trayectorias que se desea que el robot siga durante una misión. Esta información (la posición, velocidad y aceleración del robot) es enviada al Sistema de Control el cual tratará de alcanzar las referencias deseadas. El sistema de guiado tiene como entradas los Puntos Guía, definidos por el usuario, e información proveniente del sistema de Navegación. Además, puede recibir información sobre la situación del ambiente (ej. corrientes marinas), de la topografía del fondo marino, y datos provenientes de un sonar para la detección de obstáculos. No obstante, actualmente algunos AUV tienen un módulo de planificación dinámica que les permite re-planificar durante la misión y pasarle al módulo de guiado puntos guía que no fueron definidos previamente [85].
- Sistema de Control. Es el encargado de definir las fuerzas necesarias que debe generar el sistema de propulsión para que el robot alcance las referencias deseadas. La construcción del sistema de control involucra el diseño y la sintonización de las Leyes de Control que regularán los movimientos del robot. El sistema de control recibe información de la posición, velocidad y aceleración deseada y el estado actual de estas variables medidas a través de los sensores del robot. Este sistema puede consistir de leyes de control clásico, control no lineal, control inteligente, sistemas basados en comportamiento (behavior-based), etcétera [85]. Un ejemplo de ello, es el control de seguimiento de trayectoria diseñado para un vehículo marino de superficie relacionado con el presente trabajo que incluye una acción de control integral para compensar las perturbaciones contantes de las olas o el viento [86].
- Sistema de Navegación. Este sistema recibe la información de los sensores de posición, velocidad y aceleración del robot submarino. Mediante un observador u otros

algoritmos procesa esta información y posteriormente la envía al Sistema de Guiado y al Sistema de Control [85].



Figura 2-10. Diagrama simplificado de la arquitectura de software de MEDUSA [84].

Se recomiendan dos diseños de transectos para el monitoreo marino: cuadriculas anchas y cuadrículas densas. Las cuadrículas anchas que pueden cubrir líneas paralelas de entre 2000 y 4000 m son eficientes en propósitos de monitoreo de objetos grandes y en tareas de mapeo, las cuadrículas densas son adecuadas para tareas de monitoreo basadas en foto-mosaicos o búsqueda de especies u objetos pequeños [87]. Por ejemplo, para monitoreo de la estructura física del lecho marino o la biota, las cuadrículas densas son la más adecuadas, como en el trabajo del autor Ferrari y su equipo en 2016 que hicieron un monitoreo de corales en Australia [88] con este tipo de transecto.

En la Figura 2-11 se puede observar una trayectoria del robot MEDUSA, en una de sus misiones. Es claro que se trata de una planeación de trayectoria convencional, es definida por el usuario en un inicio, se intenta abarcar el mayor espacio con trayectorias paralelas, separadas por un espacio considerado por la distancia de visión dada por su sistema [84].



Figura 2-11. Esquema de una trayectoria de una misión de MEDUSA [84].

En la Figura 2-12, se muestra un ejemplo de uso de cuadrículas densas y anchas del trabajo de Morris y colaboradores; a la izquierda de la imagen realizan un monitoreo con cuadriculas densas aproximadamente (de 25 x 25 metros) de 3.57 m de separación entre líneas, en el centro se observa una cuadrícula ancha (entre los 1000 y 1500 metros de largo) y a la derecha una combinación de las cuadriculas [89].



Figura 2-12. Ejemplos de uso de monitoreo en cuadrículas densas y anchas [89].

2.2.3. Vehículos submarinos aplicados al pez león

Existen un número bajo de investigaciones reportadas sobre el uso de vehículos submarinos no tripulados enfocados a la problemática de la invasión del pez león. Específicamente, se tiene un reporte en la revista Complexity de Hindawi [55], dos reportes encontrados en internet; uno de tesis de la Universidad de Pardue de Indiana [90], y el segundo, del Instituto Politécnico de Worcester en Massachusetts [57]. En dos de los casos, se diseñaron vehículos ROVs, sólo en [90] se llegó a una versión teórica para un AUVs. Por otro lado, la empresa RSE (por Robots in Service of the Environment) lidera en la parte comercial, los desarrollos de ROVs diseñados para capturar o cazar peces león [91].

Para el primer trabajo en 2018 [55], en el departamento de ingeniería eléctrica de la Universidad de la Mar en Texas, Estados Unidos, desarrollaron un vehículo submarino operado remotamente (Figura 2-13) para realizar monitoreo del pez león. El usuario usa un Joystick para manejar e investigar por el fondo marino, conectado por un cable para envió de datos y soporte de energía.



Figura 2-13. Características del ROV (La Mar); a) diagrama de bloques de la sección en superficie y submarina, b) imagen del ROV y descripción de partes [55].

En 2014 se presentó un reporte de tesis de la universidad de Pardue, este equipo de trabajo pretende el diseño de un AUV inspirado biológicamente en un pez, sin embargo, solo llegaron a propuestas teóricas (Figura 2-14); la tarea autónoma que realizaría este vehículo se basa en seguimiento de puntos por medio de procesamiento de imágenes, además, del uso de sensores de profundidad y de proximidad [90].



Figura 2-14. Versión preliminar del vehículo submarino bio-inspirado [90].

En el trabajo del Politécnico de Worcester en 2019 [57], reportaron un prototipo de un ROV para la captura del pez león, al igual que el trabajo anterior, éste no llegó a hacer pruebas finales como se observa en la Figura 2-15. Este vehículo sería manejado remotamente desde la superficie, tendría la función de captura del pez león.



Figura 2-15. Prototipo de ROV del Politécnico de Worcester [57].

En relación a los diseños comerciales, la empresa RSE tiene un vehículo ROV llamado "*The guardian LF1*" (Figura 2-16), el cual, va dirigido a la asistencia de equipos de investigación para la captura de peces león. Usa un voltaje bajo y controlado para inmovilizar al pez antes de llevarlo dentro de un contenedor. Es manejado desde la superficie, el cable de conexión entre el vehículo y la superficie puede tener una longitud de 200 metros, este le provee de energía y comunicación para su control [91].


Figura 2-16. Vehículo ROV "The guardian LF1" [91].

Los vehículos desarrollados para combatir la problemática del pez león, hasta el momento, son robots operados remotamente, un caso intentó desarrollar un vehículo autónomo [90], pero sin llegar a su desarrollo o a una simulación de su propuesta.

2.2.4. Generación de caminos óptimos

En este apartado se pretende exponer sobre los problemas de rutas en vehículos y su importancia de reducir las distancias de trayectoria y generar rutas de manera anticipada.

Los problemas sobre la optimización de rutas tienen su origen en el siglo XVIII, inicialmente no con la finalidad de optimización, cuando los habitantes de Königsberg, un pueblo de la actual Rusia, empezaron a debatir si existía alguna ruta que pasase una única vez por los 7 puentes que atravesaban el río Pregel y volviese al punto de origen. El problema se le propuso al matemático suizo Leonhard Euler, el cual demostró que no existía ninguna ruta que cumpliera las condiciones, en un artículo del año 1736 [92].

2.2.4.1. Los problemas de rutas en vehículos

El primer problema que comprende encontrar la ruta más óptima, fue el problema del cartero chino (CPP por Chinese Postman Problem) definido en 1962 [93], el cual se busca un camino de distancia mínima que recorra todos los arcos (caminos entre puntos) de un grafo al menos una vez. Otro de los problemas de rutas es el conocido como el problema del cartero rural (RPP

por Rural Postman Problem) en el trabajo de Orloff [94] y consistió en determinar el camino de distancia mínima que se recorre en algunos de los arcos de un grafo.

La teoría de grafos pretende definir los elementos que intervienen en un problema, donde se busca encontrar un camino o un camino más corto de entre distintos puntos a visitar, pueden o no ser visitados todos los puntos o se puede o no pasar por todos los caminos. Existen dos grandes tipos de problemas de rutas según lo que se pretende resolver, si se le da prioridades a los nodos o a los arcos. En el primero de los casos, conocido como ciclo Hamiltoniano (por W. R. Hamilton), la ruta óptima a determinar debe visitar todos los nodos, mientras que el segundo, se debe recorrer todos los arcos del grafo que define el problema, estos últimos se conocen como caminos Eulerianos debido al ser Leonhard Euler el principal exponente en trabajar problemas de este tipo como el de los habitantes rusos de Königsberg [95].

En el mismo sentido de la reducción de distancias para la disminución de consumo de combustibles o energía, según WEEA [96], la Unión Europea tiene la intención de reducir las emisiones de gases de efecto invernadero en un 20% y un 40% para 2020 y 2030, respectivamente. Sbihi y Eglese [97] introdujeron el campo de investigación de la logística verde, que se ocupa de la sostenibilidad de los procesos de entrega teniendo en cuenta factores ambientales y sociales. Con la penetración del mercado de vehículos eléctricos (EV), muchas empresas de logística evaluaron el uso de los EV como parte de su flota para disminuir las emisiones.

2.2.4.2. Problema del viajante de comercio (TSP)

Es un problema común en el ciclo de Hamilton y es catalogado como un problema de complejidad *np*, es decir, el número de posibles soluciones crece exponencialmente con el número de nodos del grafo (ciudades) y rápidamente sobrepasa las capacidades de cálculo de los ordenadores más potentes [98]. Consiste básicamente en "un viajero que quiere visitar *n* ciudades una y solo una vez cada una, empezando por cualquiera de ellas y regresando al mismo lugar del que partió; suponiendo que conoce la distancia entre cualquier par de ciudades, la solución debe ser el camino que produzca la distancia mínima [99].

El origen del término "Traveling Salesman Problem" permanece aún desconocido, pues no existe documentación que apunte a ningún autor en concreto; la primera referencia a este término parece ser un artículo de 1949 de Julia Robinson [100]. "On the Hamiltonian game (a traveling salesman problem)".

25

Existe una gran variedad de aplicaciones de este problema para optimizar las rutas de empresas de trasporte y logística [101, 102, 103], es un caso particular del Problema de Rutas de Vehículos (VRP por Vehicle Routing Problem). No solo puede resolver cuál será la mejor ruta o la menor distancia a recorrer, la versatilidad de este tipo de operaciones combinacionales permiten agregar al modelo a evaluar una gran variedad de restricciones o características, por ejemplo, existen trabajos donde se agrega al análisis el costo del combustible [104, 105, 106], en vehículos eléctricos el consumo de energía y estado de carga de la batería [107, 108, 109] incluso penalizaciones de tiempo o por uso de combustible [110, 111]. Los objetivos de aplicar este tipo soluciones a los problemas de rutas, es la disminución de la contaminación, un consumo menor de combustible o energía y un ahorro considerable en los costos de operación.

2.3. LIMITACIONES DEL ESTADO DEL ARTE

- Los trabajos que se dedican a resolver el reconocimiento del pez león tienen resultados no satisfactorios (menores al 32% de precisión) [27] o no están completos como sistemas de reconocimiento soló como clasificador [56] o un foto-identificador [54], finalmente, se tiene un sistema con porcentajes del 92% de precisión, sin embargo, carece de un análisis detallado de las métricas y el procedimiento para justificar como válido el resultado.
- Los métodos tradicionales de UVC realizados por buzos, se realizan durante el día, subestimando la abundancia del pez león durante la noche.
- Las trayectorias de los AUVs son establecidas por el usuario (no se calculan automáticamente), pueden ser lineales, en zigzag (celdas rectangulares o cuadrículas) o espiral, depende de la tarea a realizar. Por lo que no existen trabajos de generación de rutas automáticas para los AUVs, sólo investigaciones que buscan hacer más eficiente la trayectoria de un punto a otro.
- Los AUVs comerciales y de investigación realizan levantamiento de información durante el monitoreo para observar de manera remota o procesar posteriormente, no monitoreo con detección automática en tiempo real.

CAPITULO 3

METODOLOGÍA DE LA GENERACIÓN AUTÓNOMA DE TRAYECTORIAS

Dentro de los distintos retos de ingeniería que presenta el desarrollo del vehículo submarino propuesto en el proyecto 2015-01-786 de CONACYT, este trabajo de investigación se propone resolver, en base a visión artificial; la identificación del pez león, además de dar autonomía al vehículo submarino, en cuanto a la toma de decisiones de las trayectorias en busca del pez.

La metodología de este trabajo de investigación se desarrolla en el siguiente orden:

- Planeación de rutas óptimas para minimizar el consumo de energía del sistema de almacenamiento interno de energía.
- Diseño del sistema de reconocimiento para la identificación del pez león.

En este capítulo se presenta la metodología para la generación de trayectorias óptimas y el sistema de reconocimiento del pez león es presentado en el siguiente capítulo.





La generación autónoma de trayectorias se realizará desde un punto superior, aproximadamente a 8 metros de altura del fondo marino para este sistema propuesto de visión estereoscópica (puede ser mayor distancia para otros métodos de determinación de profundidad), al terminar este proceso enviará las coordenadas al sistema de control para el movimiento del AUV (o un segundo AUV) para que realice el reconocimiento del pez león, se encargue de la orientación y velocidades adecuadas para desplazarse a cada punto (no visto en este trabajo). Al llegar el vehículo a cada sitio de interés, podría detenerse unos segundos para realizar el reconocimiento de peces (Figura 3-1), además, de hacerlo también durante el

desplazamiento, finalmente, posicionarse en el área contigua para repetir el procedimiento. En este trabajo no se define si habrá un solo AUV o dos, solamente se describe el proceso propuesto de generación de trayectorias.

Este proceso deberá repetirse el número de veces necesario para cubrir un área total requerida por el usuario. Se observa en la Figura 3-2 una propuesta para un área compuesta de 16 subáreas de censado, pueden ser más o menos y serán determinadas por el usuario. El proceso de censado y obtención de trayectorias se realiza en los puntos centrales anaranjados desde un lugar elevado, las flechas anaranjadas señalan la ruta del vehículo que realizará este censado, los puntos verde y rojo en las áreas indican el inicio y fin del monitoreo, respectivamente, realizado por el mismo vehículo que hace el censado o un AUV distinto.

En una etapa posterior a este trabajo, se puede determinar números diferentes de áreas y distintas configuraciones del orden para censarlas y realizar el reconocimiento, para este caso en particular, consiste en la suposición de 16 sub áreas y un ejemplo de usar dos AUVs y otro ejemplo de usar sólo un AUV, explicados a continuación:

- Trayectoria de censado por un vehículo distinto (Figura 3-2a): El vehículo comienza a censar en el sub-área 1 en el orden numérico, finalizando en la 16, justo cuando pase el sub-área 1 y obtenga su trayectoria, enviará la información del orden de las coordenadas (trayectoria óptima) al segundo AUV para comenzar la tarea de reconocimiento a nivel del fondo marino, así sucesivamente, finalizando también en la sub-área 16, para comenzar con otro bloque.
- Trayectoria de censado por el mismo vehículo de monitoreo (Figura 3-2b); al ser un vehículo único, una opción de ruta a seguir para el censado es empezar en la 1 y terminar en un área contigua (en este caso la 8), al finalizar de censar, puede comenzar con el monitoreo en las rutas óptimas encontradas en cada área en el orden numérico u otro orden, pero finalizando en una de las áreas que permita iniciar el censado en un nuevo bloque.



Figura 3-2. Ejemplos para la trayectoria de censado del AUV; a) trayectoria para un vehículo distinto al de monitoreo, b) trayectoria para el mismo vehículo que realiza el monitoreo.

El o los vehículos deberán tener un dispositivo que les dé información de las coordenadas latitud, longitud y profundidad, además de la comunicación entre ellos (no visto en este trabajo).

3.1. EQUIPO UTILIZADO

Es necesario mencionar que la metodología, en este punto, cambió debido a la pandemia global de la enfermedad Covid-19, de pasar a realizar pruebas reales en campo a hacerlo de manera virtual, en una simulación. Para esto, se ha elegido un entorno de trabajo para robots que provee la funcionalidad de un sistema operativo (ROS) el cual se integra con el simulador Gazebo, para simular un sistema de visión estéreo dentro un vehículo submarino, un espacio marino, arena, iluminación y turbidez, así como, distintas estructuras marinas como rocas, arrecifes, pastos, etcétera. Se enlista a continuación el equipo usado para este sistema:

Hardware

- Jetson Nano Nvidia: 4GB de RAM, Procesador Quad-core ARM A57 a 1.43 GHz y un GPU integrado Tegra X1 149-core.
- Memoria SD 64GB, para montar el sistema operativo.
- Pantalla LCD.
- Cables para conexiones.

Software

- Blender 2.90.0; para crear elementos 3D (piedras, pastos, arrecifes, corales) uso en Windows.
- Ubuntu Linux 20.04 LTS (adaptada a Jetson Nano).
- ROS versión Noetic Ninjemys [112].
- Simulador multi-robot Gazebo 11.8.1[113].
- OpenCV 3; librería para hacer uso de funciones de visión por computadora [114].
- Librería CPLEX de IBM; para obtención de caminos óptimos entre puntos [115].
- Paquetes ROS:
 - o gazebo_ros_pkgs; para la configuración adecuada entre ROS y Gazebo.
 - uuv_simulator (adaptada a versión Noetic); para hacer uso de elementos ambientales del simulador (niebla, nubes, superficie y malla del fondo).
 - o cv.bridge; para leer información de cámaras simuladas con OpenCV.
 - ros_essentials_cpp; ayuda a las configuraciones de cv_bridge con OpenCV y a la de emisión y recepción de datos de las cámaras entre nodos.

3.2. DISEÑO DEL SISTEMA DE GENERACIÓN AUTÓNOMA DE TRAYECTORIAS

La solución a partir de visión artificial es necesaria implementar un sistema de generación de rutas óptimas para la exploración submarina. Básicamente, se propone un proceso en el que el AUV realizará la búsqueda siguiendo el proceso que haría un humano para buscar un objeto en una determinada área; observa, localiza lugares probables, prioriza al ordenar los sitios, se desplaza y realiza la búsqueda en cada punto.



Figura 3-3. Diagrama del sistema de generación autónoma de trayectorias.

La propuesta general para obtener trayectorias óptimas de manera autónoma por medio de visión artificial, involucra resolver las siguientes etapas (ver Figura 3-3):

- Construir un espacio en 3D con características visuales similares a un fondo marino; incluyendo las estructuras marinas y un vehículo con cámaras virtuales para la visión estereoscópica.
- Diseñar un sistema de detección de estructuras marinas (ver su profundidad) mediante visión estereoscópica; esto en función de las estructuras frecuentadas por el pez león, a partir de una imagen adquirida desde un punto superior.
- 3. Encontrar puntos de interés mediante técnicas de PDI alrededor de las estructuras y sobre las estructuras más amplias a partir de la imagen de profundidad.
- 4. Realizar corrección geométrica de las coordenadas en pixeles a distancias en metros de los puntos de interés encontrado.
- 5. Generar la ruta más corta que pase por todos los puntos de interés, por medio de TSP.

3.2.1. Diseño del espacio marino en 3D

Antes de comenzar con los procesos de generación de trayectorias óptimas, se presenta y explica de manera general los espacios y elementos involucrados en la simulación del ambiente marino.



Figura 3-4. Elementos 3D de vegetación simulados en Blender

En el software Blender se crearon distintos elementos en tercera dimensión, como vegetación (Figura 3-4), rocas, arrecifes y corales (Figura 3-5) guardados en el formato ".dae" aceptado por Gazebo.



Figura 3-5. Elementos de rocas, arrecifes y corales simulados en Blender.

Se configuró un espacio en Gazebo conformado por cuatro muros, la base (fondo marino) y la tapa o techo (superficie del agua), con sus dimensiones y características visuales. Dentro del espacio simulado se colocaron los distintos elementos, cambiando las posiciones y tamaños para generar distintos escenarios. En la superficie se simuló el AUV, sólo teniendo una característica funcional, la visión estereoscópica (Figura 3-6).



Figura 3-6. Configuraciones de espacio simulado en Gazebo.

La interacción de los nodos en el sistema ROS durante la simulación se presenta en la Figura 3-7, donde el ovalo en rojo (/gazebo) representa el nodo maestro y el ovalo negro (/image_converter) representa el nodo secundario que recibe los mensajes (datos en matrices) de las cámaras derecha e izquierda.



Figura 3-7. Configuración gráfica de los nodos en el sistema ROS.

3.2.2. Diseño del sistema de detección de estructuras marinas

Este sistema deberá ejecutarse desde una parte superior del fondo marino, es el primer paso para identificar lugares de interés (estructuras marinas) donde el pez león suele estar, en consecuencia, es necesario conocer las características visuales que tienen vistas desde la parte superior a diferentes alturas. En la Figura 3-8, se pueden observar algunas configuraciones de espacio en el ambiente marino simulado, de 42 creadas. Estos 42 escenarios son clasificados en 3 distintos grupos, 14 escenarios por grupo, dependiendo del número de elementos ocupando el espacio simulado; alta densidad, media densidad y baja densidad.



Figura 3-8. Ejemplos de escenarios simulados en Gazebo.

Investigadores ligados al presente trabajo, de la Unidad de Ciencias del Agua del CICY, quienes realizan trabajos de monitoreo espaciales del pez león en el Caribe mexicano, compartieron datos de los lugares donde frecuentemente se encuentran a los peces, además, de los reportes de investigación de monitoreo en distintas publicaciones [8, 9,61, 67]. Los peces león se encuentran en:

- Sobre pastos marinos en las oquedades existentes entre el fondo arenoso y los manchones de pasto.
- Sobre corales pétreos (coral rocoso muerto) sobre todo en los que están llenos de espacios disponibles para encuevarse.
- Sobre la barrera arrecifal, ya sea entre las oquedades o sobre la cima o cresta de los corales (sobre todo los corales constructores cuerno de alce y cuerno de ciervo).
- En las raíces de los manglares y en las cuevas o galerías que se forman entre las raíces.
- En las bases de los corales blancos (abanicos) protegidos en ellos.
- Debajo de los muelles en las maderas de soporte y en embarcaciones hundidas.

La información anterior nos permite definir la estrategia para localizar puntos (puntos de interés) donde el vehículo deba visitar para buscar al pez león.

3.2.2.1. Visión estéreo del fondo simulado

En este apartado se describe brevemente el primer paso de procesamiento de imágenes realizado a los espacios simulados, la visión estereoscópica. Se han configurado las cámaras virtuales con una separación de 30 cm entre ellas, una apertura horizontal y vertical de 80°, una resolución de 2464x2464 pixeles (permitida para el modelo IMX219-120), a una altura de 8 metros (Figura 3-9). Esta configuración permite tener una superficie del fondo marino de 12.8 m horizontales y 13.4 m verticales de censado.



Figura 3-9. Configuración óptica de la visión estéreo.

Inicia el proceso capturando dos imágenes (izquierda y derecha) desde las cámaras virtuales que se encuentran debajo del AUV, en dirección hacia el fondo marino, Figura 3-10a. Enseguida, se aplica la función "cv2.StereoBM_create" de la librería OpenCV para configurar el número de disparidades y el tamaño de bloque de los pixeles a comparar entre imágenes, luego se usa la función "configuracion.compute" para procesar ambas imágenes, lo cual da como resultante una imagen de profundidad, una matriz a escala de grises cuyos pixeles contienen valores de profundidad (Figura 3-10b). Los pixeles marcados como blancos indican un cambio de profundidad detectada (mínimo de 30 cm aproximadamente), mientras las áreas estén más oscuras son objetos o áreas más altas (más cercanas a la cámara).



Figura 3-10. Imagen de profundidad a partir de 2 imágenes por visión estéreo.

En este punto ya se han detectado las estructuras marinas del espacio simulado, representadas por los mismos pixeles blancos, distinguiéndose como contornos o pequeñas áreas de las estructuras. En el anexo 1 se encuentra el código de programación para el cómputo de la visión estereoscópica.

3.2.3. Localización de los puntos de búsqueda

Teniendo en cuenta los lugares de preferencia del pez león vistos en la sección 3.2.2, de la cual se concluye, para esta investigación, una búsqueda alrededor de las estructuras marinas y sobre ellas. Es decir, los puntos de interés deben estar colocados espaciadamente (entre 2 y 3 m) alrededor de las estructuras a aproximadamente 1m de distancia (para mayor probabilidad de detección del pez en lugares con baja iluminación) y puntos encima de estructuras mayores a 30 cm de anchura, menores a esa medida es posible que no se puedan generar puntos encima o alrededor, por la baja probabilidad de encontrar los peces león, además, de que hay más posibilidad de visualizarlos encima cuando se busca alrededor de esas estructuras pequeñas.

Partiendo de la imagen en escala de grises de la Figura 3-10b donde se localizan las estructuras detectadas y siguiendo las condiciones de localización de puntos, se plantea el

siguiente procedimiento para el procesamiento de la imagen, usando las distintas funciones de la librería OpenCV:

- Eliminar zonas externas blancas: para no procesarlas como estructuras, se originaron en el procesamiento estéreo al no ser zonas de coincidencia entre las imágenes izquierda y derecha.
- 2. Dilatar zonas blancas: aproximadamente 1m interna y externamente, es la distancia que definida con anterioridad para colocar los puntos sobre y alrededor las estructuras.
- Encontrar contornos: Los contornos de estos espacios dilatados marcan las zonas (alrededor e internamente de las estructuras) donde pueden colocarse los puntos de interés.
- 4. Localizar contornos pequeños: si éstos tienen su centro negro en la subimagen dilatada (contornos internos de la dilatación) será puntos de interés.
- 5. Definir líneas verticales y horizontales: pueden ser 3 posibilidades; 3 líneas, 4 líneas y 5 líneas verticales y horizontales.
- 6. Encontrar intersecciones líneas-contornos: los puntos de intersección son puntos de interés.
- 7. Intersección entre líneas verticales y horizontales: si se elige monitoreo completo; si los puntos de intersección de las líneas son negros en la subimagen dilatada, son puntos de interés, en cambio, si el monitoreo elegido es libre, no se realiza esta acción.
- 8. Promediar puntos cercanos: de los puntos de interés ya encontrados en los pasos anteriores, se buscan los más cercanos entre sí (menores a 350 pixeles o 1.95m aproximadamente), se promedia su distancia y se genera el punto promedio. Estos puntos promedio y los que no se procesaron por no ser cercanos, son los puntos de interés finales.

Se ha programado este proceso para que el usuario pueda elegir entre dos tipos de monitoreo; completo y libre. El primero se refiere a abarcar puntos establecidos a ciertas distancias además de los puntos alrededor de las estructuras encontradas y el monitoreo libre a sólo alrededor de las estructuras localizadas. Este último es la propuesta principal para esta investigación. Además, con fines de prueba, se establecieron tres distintos números de líneas de intersección (Figura 3-11). Estas líneas al interceptar con los contornos de las estructuras, generan un número de puntos dependiendo de la cantidad de líneas. En este sentido, la opción de 3 líneas de intersección (Figura 3-11a) tienen una distancia aproximada entre ellas de 3.25 m, es decir, puede haber puntos de interés a esa distancias de separación, por otro lado, para

38

la opción de 5 líneas puede haber puntos separados a 2.16 m. Distancias más pequeñas no son deseables, los puntos estarían muy cerca entre sí, aunque se detecten estructuras pequeñas, al final se promedian los puntos cercanos, no son puntos tan relevantes. Mientras más líneas, más probabilidades de interceptar estructuras pequeñas, pero habrá puntos muy cercanos entre sí, por el contrario, si hay pocas líneas, menos estructuras interceptadas, puntos muy alejados entre sí, por consecuencia, espacios sin monitorear (sesgos). De esta forma se eligió probar con 3, 4 y 5 líneas, para definir cuál de ellas tiene mejor desempeño.





Cada uno de los pasos a seguir en esta etapa, se muestra en código en el anexo 2, incluyendo las opciones a elegir por el usuario de las líneas de intersección y el modo de monitoreo.

3.2.4. Transformación de coordenadas de pixeles a metros

El objetivo en esta etapa es determinar las coordenadas (en metros) de los puntos de interés del área del fondo marino que se encuentran en coordenadas de pixeles en una imagen de 2464x2464. Si tomamos en cuenta los pixeles vertical y horizontal de la cámara, que en realidad son los grados de apertura, en este caso particular son 80° de apertura para ambos casos (Figura 3-12).



Figura 3-12. Apertura de los ángulos en una cámara.

Observando de costado la apertura horizontal de la cámara obtenemos la Figura 3-13a. En ella podemos observar triángulos rectángulos que relacionan las distancias involucradas de cualquier punto en el fondo marino.



Figura 3-13. Elementos que se relacionan en el arreglo óptico; a) corte horizontal y b) triángulo generado hacia un punto de interés.

En la Figura 3-13b, el triángulo rojo; " β " es la apertura horizontal en pixeles de la cámara en el punto de interés (cuadro verde), "a" es la profundidad del vehículo al fondo marino y "c" es la distancia del vehículo al punto de interés (hipotenusa del triángulo).

Por ejemplo, para la parte horizontal se tienen 2464 pixeles, la razón de grados en cada pixel corresponde a (gP: *grados de pixel*):

$$1 \ pixel = \frac{^{\circ}CamaraH}{2464} = gP \tag{1}$$

Donde °*CamaraH* son los grados horizontales de la cámara. Para encontrar los valores de " β ", es necesario que sean dependientes de la apertura de pixeles (Ap) que va desde el centro de la imagen (1232,1232) hasta el punto de interés (coord_x, coord_y);

$$Ap = \operatorname{coord}_{\mathbf{x}} - 1232,\tag{2}$$

$$\beta = (Ap)(gP),\tag{3}$$

La función tangente relaciona los lados "b" y "a", los valores conocidos serán " β " y "a" en la ecuación 4:

$$\tan\beta = \frac{Op}{Ady} = \frac{b}{a} \tag{4}$$

$$b = (\tan \beta)(a) \tag{5}$$

Despejando "b" obtenemos la ecuación 5, este resultado es la distancia (en metros) del centro de la imagen al punto de interés en la apertura horizontal. Este mismo procedimiento es realizado para la apertura vertical. Luego de aplicar este procedimiento a las coordenadas de puntos, se obtiene la transformación de metros (Figura 3-14).



Figura 3-14. Transformación de coordenadas de pixeles a metros.

Luego de este proceso, con las medidas de localización en metros de los puntos de interés en el fondo marino, se realizará el proceso de generación de trayectoria óptima pasando por las coordenadas encontradas. El código de este proceso se encuentra en el anexo 3.

3.2.5. Generación de ruta más corta

En este trabajo resolveremos el problema de rutas por nodos, en particular en el problema del viajante de comercio (TSP, por Travelling Salesman Problem), históricamente se entiende como un caso particular del problema de rutas de vehículos (VRP, por Vehicle Routing Problem), en el que se trata de averiguar las rutas óptimas de una flota de transporte para dar servicio a clientes.

Dantzing [116] desarrollo el método del "simplex" en 1947 para resolver problemas de programación lineal, y hacia 1953, existían adaptaciones del mismo para casos especiales de transporte. Gracias a estos desarrollos, en la actualidad tenemos un avance importante como las librerías de IBM ILOG CPLEX que vienen disponibles para C, C++, Java, .NET y Python que solucionan problemas de programación lineal (LP) y relacionados. Específicamente, resuelve de forma lineal o cuadrática problemas de optimización con restricciones donde el objetivo a optimizar se puede expresar como una función lineal o una función cuadrática convexa. Las variables del modelo se pueden declarar como continuas o como más restringidas para tomar solo valores enteros [117]. Existe una gran cantidad de trabajos para la solución de TSP o de VRP que se apoyan en las librerías CPLEX [118, 119, 120, 121, 122]

Hasta este momento se tienen los puntos de interés o sitios importantes que se desean visitar en el fondo marino. Esta etapa genera una ruta que se establece como óptima, tomando en cuenta todos los puntos, además del punto inicial y final, establecidos en la propuesta de censado vistos en la Figura 3-2, el cual sus posiciones dependen del espacio a llegar para comenzar una nueva búsqueda. En nuestro caso, todos los escenarios serán evaluados con el punto inicial en la parte inferior de la imagen y el final en el otro extremo (parte superior).

3.2.5.1. Implementación y solución del TSP

Teniendo la información de los arcos entre todos los puntos visualizada en dos dimensiones, se procede a la aplicación de un mecanismo de solución a este tipo de problemas (TSP). Se usará una librería de IBM llamada Cplex, a la que se hizo referencia en el capítulo.

Para iniciar con la aplicación de este método analítico de programación lineal, necesitamos definir los elementos que se usarán; definimos V como un conjunto de todos los vértices (puntos) que se desean visitar, con subíndices v_i y v_j que indican el vértice de partida y el de llegada, respectivamente. La variable de decisión binaria x_{ij} , el cual debe tener un par de valores sólo de 0 y 1, por ejemplo $x_{ij}=(1,1)$ esta configuración significa que el vértice de partida *i* y el de llegada *j* están participando generando un arco o camino entre ambos vértices, si uno o ambos valores fueran cero, no habría una conexión entre ellos, además, tenemos a D como el conjunto de arcos formados entre los vértices *i* y el vértice *j*, donde $i \neq j$, siendo d_{ij} la distancia entre el vértice de partida (v_i) y el vértice de llegada (v_i).

Se va a utilizar una versión generalizada del TSP adaptado a nuestro problema, tomando en cuenta que definimos un punto inicial y un punto final, es decir, no necesitamos que el recorrido cierre el ciclo, requerimos un grafo dirigido, por esa razón, es necesario definir ciertas condiciones. En primer lugar, es necesario definir que todos los vértices se conectan entre sí, con algunas restricciones;

$$\sum_{\forall i \in V} x_{i,j} = 1$$

$$\sum_{\forall j \in V} x_{i,j} = 1$$
Sólo si j ≠ 0; 0 = vértice inicial (7)
Sólo si j ≠ 0; 0 = vértice inicial (7)

La ecuación 6 permite que puedan salir de todos los vértices enumerados con el subíndice *i*, pero cuando pretenda salir del vértice final (v_n), ya no estarán disponibles todos los demás, en cambio, la ecuación 7 nos ayuda a visitar todos los vértices enumerados con el subíndice *j*, siempre que el vértice inicial (v_0) no sea un punto de llegada. Con estas condiciones se logra visitar todos los vértices eligiendo uno inicial y uno final, sin tener retorno o cerrar el recorrido.

Adicionalmente, se debe definir una función objetivo, que es encontrar la mínima distancia de la suma de los recorridos, la cual se expresa;

$$Min Z = \sum_{\forall D} x_{ij} d_{ij} \tag{8}$$

Teniendo todos los datos, ecuaciones y restricciones, así como, la programación propia de *python* y de la librería de Cplex, se corre el algoritmo de TSP, brindando la solución óptima o cercana a la óptima. El algoritmo de solución más común que resuelve el problema es el llamado *branach&cut*, debido a que tiene excelentes resultados para un número promedio y alto de vértices (puntos) [123]. El código de programación se incluye en el anexo 3.

Un ejemplo de obtención de ruta óptima se observa en la Figura 3-15, para el escenario "10"; sólo hay una estructura mediana, 5 líneas de intersección para un monitoreo completo en el inciso *a*) y monitoreo libre en el *b*), además se muestra las distancias de punto a punto y la distancia total (sin contemplar la primera distancia y la última). Se puede observar en el monitoreo libre una menor distancia de recorrido.





Hasta este punto se concluye la propuesta en su primera fase. Se logra obtener trayectorias de forma automática en un ambiente simulado, para que el vehículo encargado del reconocimiento del pez león, se desplace por la ruta generada en el área o bloque de áreas censadas. Se muestran y discuten los resultados en los siguientes apartados.

3.2.6. Método de evaluación del sistema generación autónoma de trayectorias

Al ser diferente de cualquier otro sistema de monitoreo, esta propuesta de sistema de generación de trayectorias óptimas, necesita un diseño de método para evaluar su eficiencia. Si bien el objetivo de esta propuesta es obtener una menor distancia de recorrido, comparada con una ruta preestablecida (para los AUV), es necesario primero, conocer el mejor modelo de líneas de intersección.

3.2.6.1. Procedimiento para evaluar los modelos de líneas de intersección

Para esto, se ha diseñado un método manual basado en imágenes del escenario simulado, en el que se deduce qué tan bien se cubren las áreas que son importantes monitorear, encontrando sesgos o áreas que no son cubiertas por el área de visión de la cámara (en trabajos futuros se hará la evaluación mediante una simulación).

Se necesitará:

- La imagen (izquierda o derecha) del escenario utilizado para generar la visión estéreo.
- Las coordenadas de imagen de los puntos de interés.
- La trayectoria obtenida por TSP.
- Tener en cuenta las alturas de los elementos simulados (estructuras marinas).
- Librerías de OpenCV y numpy para procesar imágenes con Python (opcional).
- Software de dibujo (Paint 3D en nuestro caso).

Antes de comenzar el procedimiento, supondremos el arreglo óptico del sistema de visión de monitoreo (sistema de reconocimiento del pez león) para establecer la superficie o área de alcance de censado. Las recomendaciones de la anchura del transecto para un censo visual tradicional realizado por buzos, en algunas casos ayudados con cámaras, es de 1 a 5 metros, depende de la confianza que se tenga para detectar al objetivo con la vista o mediante cámaras [66]. Son pocos los estudios sobre reconocimiento del pez león, como se menciona en las limitaciones del estado del arte, no existen aún estudios para definir la distancia de detección del pez león mediante cámaras, sin estos, sólo queda tener suposiciones de distancia en base al tamaño del pez león, la resolución de la cámara y la eficiencia del sistema de reconocimiento. En el trabajo de Morris y su equipo [89], mencionado con anterioridad, tienen un ancho de transecto de 3.57m para una tarea de monitoreo mediante la creación de un mosaico de imágenes, es decir, será el ojo humano quien finalmente analice o detecte la especie buscada.

Teniendo en cuenta lo anterior, se decidió nuestra propuesta de arreglo óptico para el sistema de monitoreo o reconocimiento del pez león, el cual nos ayuda a definir la anchura del transecto lineal a seguir por el AUV para nuestra propuesta y para la ruta preestablecida.

Teniendo en cuenta una de las cámaras IMX219-120 utilizadas para la visión estéreo, pero ahora en 90° de apertura, en la misma posición apuntando hacia abajo, a una distancia de 1.75 m sobre el fondo marino, generará un ancho de visión sobre el fondo de 3.5 m. Con esta arreglo de la cámara, cualquier objeto que aparezca en la imagen estará a menos de 2.4 m de distancia, si se desea tener más grados de visión, sobre todo para observar las paredes de las estructuras altas, se cambiaría la configuración a 120° cuando se acerque a estos, teniendo en la imagen objetos hasta 2.92 metros de distancia como máximo (ver Figura 3-16).



Figura 3-16. Propuesta de sistema de visión para realizar monitoreo.

Teniendo en cuenta los elementos necesarios para realizar la evaluación, seguimos los siguientes pasos:

- 1. En software de dibujo se coloca la imagen del escenario a evaluar,
- 2. Encima de la imagen colocar los puntos de interés,
- 3. Dibujar la ruta óptima para este escenario; hasta este punto se podría realizar lo anterior mediante procesamiento con OpenCV.
- 4. Dibujar con un marcador grueso (3.5 m en relación a las medidas del escenario) y con transparencia; esta marca el área de visión supuesta de la cámara.

- Señalar con rojo (sin transparencia) las áreas que la cámara se supone no alcanzaría a observar (puntos ciegos o sesgos), debido a una distancia mayor a 3.5 m o espacios detrás de estructuras.
- 6. Finalmente, se contabiliza como un sesgo si una o varias pueden eliminarse desde un solo punto de vista de la cámara.

Este procedimiento será utilizado para evaluar y elegir el mejor de los tres modelos de líneas de intersección, el modelo con mejor desempeño se usará para evaluar y comparar con el modelo de ruta preestablecida mediante comparación de distancias recorridas.

3.2.6.2. Procedimiento para evaluar los modelos propuestos con la ruta preestablecida

Para una ruta preestablecida propuesta en este trabajo, un vehículo recorre la misma distancia para cualquier escenario, por lo tanto, no se requieren de análisis estadísticos complejos, sólo se comparará la distancia que de cada modelo con la única de la ruta preestablecida. Además, se comparará entre modelos y escenarios con distinta densidad de elementos. Para esto, es necesario obtener la distancia total obtenida por la ruta óptima de la solución TSP. Para obtener la distancia de los modelos, de monitoreo libre como de monitoreo meticuloso o amplio, se usará código de programación con las librerías numpy y matplotlib mediante Python.

La ruta predefinida se establece en base a las rutas convencionales vistas en la sección 2.2.2.3 que son líneas paralelas o también llamadas cuadrículas densas como en distintos trabajos de monitoreo detallado vistos en [85, 87, 88, 89]. Al suponer un sistema de visión (arreglo óptico) como el propuesto en la Figura 3-16, cuyo ancho de transecto es de 3.5m. Lo anterior nos ayuda a definir la ruta que seguiría un AUV de ruta preestablecida en busca del pez león igualando la condición del área que nuestro sistema puede mapear de 12.8 x 13.4 metros, además, del ancho del transecto que debe ser menor al transecto de visión (3.5m) para evitar áreas sin alcance visual de la cámara.

En la Figura 3-17, se observa la ruta predefinida con un ancho de transecto aproximado de 3.35m, es decir un ancho de visión de 1.67m de cada lado de las líneas o transectos. Con una distancia total de recorrido de 77.4 metros.





De la misma manera en la cual se programó la imagen de la figura anterior, se programan las imágenes de resultados que se obtienen de la generación autónoma de trayectorias óptimas de nuestro sistema. Finalmente, se comparan las distancias de los modelos de monitoreo libre, monitoreo extenso y la ruta preestablecida. En el anexo 4 se muestran los 42 escenarios con la ruta dibujada y la distancia total.

3.3. RESULTADOS Y DISCUSIONES DEL CAPÍTULO

En este apartado se muestran los productos de la generación de trayectorias óptimas. Tenemos una propuesta principal que llamamos monitoreo libre o "Mol-AUV" y otra secundaria como el monitoreo meticuloso o "Mom-AUV", las cuáles ambas podrían integrarse en el sistema final del AUV para que el usuario elija el modo de operación.

3.3.1. Resultados del sistema de generación autónoma de rutas óptimas

Cada una de las 2 propuestas de monitoreo se prueban con 3 modelos de líneas (3, 4 y 5 líneas de intersección), con 42 escenarios simulados.

Al ser 6 modelos diferentes en 42 escenarios, tenemos 252 pruebas. Sin embargo, las líneas de intersección se deben evaluar primero para elegir la que brinde menores áreas de sesgo

alrededor de las estructuras censadas, por tal motivo, para fines prácticos, mostraremos algunos resultados de ejemplo para la deducción y elección del mejor modelo de líneas para el Mol-AUV y para Mom-AUV, basándonos en las áreas con sesgo que producen. Se dan dos ejemplos de cada tipo de escenario definido (alta, media y baja densidad). Las evaluaciones se realizan siguiendo el procedimiento propuesto en la sección 3.2.5.1;





En el escenario 1 (parte superior de la Figura 3-18), específicamente para las líneas 3 y 4, se puede observar en rojo pequeñas áreas que definimos como sesgos alrededor de las estructuras que se deducen luego de dibujar manualmente el recorrido y el área que puede abarcar la visión de la cámara (3.5 m aproximadamente). El sesgo se produce en las partes inferiores y orillas de las rocas o arrecifes. Para el escenario 40, sólo se observa un pequeño punto rojo para el modelo de 5 líneas.

En la parte de superior de la Figura 3-19 se observa el escenario 5 con elementos de rocas grandes, el cual corresponde a escenarios de densidad media, se pueden observar varias áreas de sesgo, con 3 de ellos en el modelo de 3 líneas, 2 en el modelo de 4 líneas y uno para el de 5 líneas. Con la misma densidad media para el escenario 34 con pastos, no se generaron sesgos.



Figura 3-19. Sesgos generados por las líneas de intersección en escenarios de media densidad de elementos. En la Figura 3-20 se observan escenarios de densidad baja; arriba, el escenario 15 con estructuras de tamaño pequeño y abajo con un pequeño espacio de pasto, el cual generó un solo punto de interés.



Figura 3-20. Sesgos generados por las líneas de intersección en escenarios de baja densidad de elementos. Después de observar el comportamiento de las líneas de intersección en los escenarios de las 3 figuras anteriores, se concluye que el modelo de 5 líneas genera menos sesgos, esto se

puede observar mejor cuando se censan escenarios con estructuras grandes (escenarios 1 y 5), ya que para estructuras poco altas y poco anchas no hay mucha diferencia entre los modelos (no se generan muchos sesgos en general).

Ahora bien, aunque los puntos generados alrededor de las estructuras con el mismo número de líneas, son los mismos en los modelos Mol-AUV y en Mom-AUV, existen otros puntos en el segundo modelo que pueden hacen la ruta más larga.

Tabla 3-1. Resultados de distancias de rutas óptimas para los escenarios; separados por baja, media y alta densidad.

Escenario	Densidad del escenario	Distancia de ruta (m)		enario	sidad del cenario	Distancia de ruta (m)		enario	sidad del senario	Distancia de ruta (m)	
		Mom-AUV	Mol-AUV	Esc	Denses	Mom-AUV	Mol-AUV	Esc	Den: es(Mom-AUV	Mol-AUV
6		64.27	27.03	3		64.04	47.94	1		66.29	64.31
7		63.58	27.38	4		58.57	39.12	2		68.93	53.42
8		55.50	14.64	5		60.65	43.56	19		60.46	28.94
9		54.16	20.57	16		57.62	20.68	20		65.76	57.26
10		57.39	18.74	17		57.15	26.64	21		58.31	49.20
11		55.12	13.63	18		56.02	30.21	28		66.35	60.61
12	Poio	56.80	15.37	22	Madia	57.11	36.47	29	Alta	62.77	49.55
13	Баја	58.94	16.31	23	wedia	56.36	29.00	30		64.08	45.12
14		59.60	16.57	24		59.35	44.99	31		66.83	64.11
15		55.22	21.67	25		66.81	44.22	32		62.66	52.32
27		60.38	18.15	26		55.63	34.86	38		69.65	67.32
33		64.04	22.69	34		60.08	40.12	39		61.65	62.57
36		55.19	13.40	35		54.97	36.11	40		53.76	44.38
42	E E E E E E E E E E E E E E E E E E E	64.38	34.56	37		61.12	38.07	41		72.31	52.54

En la Tabla 3-1, se muestran los resultados de la distancia total de la ruta autónoma obtenida en cada escenario de los modelos Mom-AUV y Mol-AUV de 5 líneas; del lado izquierdo, presenta los escenarios de baja densidad de elementos o estructuras marinas, en la parte central, los escenarios de media densidad y, a la derecha, los escenarios de alta densidad. La distancia de la ruta preestablecida es de 77.4m (no presentada en la tabla por ser única).

En la Figura 3-21 se puede observar en imágenes el escenario 6; definido como un espacio de densidad baja de estructuras, a la izquierda se encuentra la representación de la profundidad y las diferencias de altura en color guinda de la estructura, a la derecha los modelos de monitoreo con las trayectorias óptimas resultantes (arriba) y su distancia total (excepto la primera y última distancia). Este escenario está compuesto por pastos cortos no mayores a 30 cm.



Figura 3-21. Resultados del monitoreo libre (Mol-AUV) y minucioso (Mom-AUV) sobre el escenario 6 de densidad baja de elementos.

Para la Figura 3-22 se tiene un escenario de densidad media de elementos, conformado por estructuras de arrecifes y corales. Finalmente, un escenario de densidad alta en la Figura 3-23, está formado en su mayoría por rocas y un pequeño arrecife, se logra observar una roca alta en la parte izquierda y debajo de la imagen de profundidad, debido a su coloración azul oscura.



Figura 3-22. Resultados del monitoreo libre (Mol-AUV) y minucioso (Mom-AUV) sobre el escenario 23 de densidad media de elementos.



Figura 3-23. Resultados del monitoreo libre (Mol-AUV) y minucioso (Mom-AUV) sobre el escenario 1 de densidad alta de elementos.

Se puede observar, en los resultados, que los modelos propuestos tienen distancias de recorrido menores a la ruta preestablecida, Así también, diferencias entre los dos modelos propuestos Mol-AUV y Mom-AUV. En los siguientes apartados se discutirá y concluirá sobre ello.

3.3.2. Análisis y discusiones

A partir de la Tabla 3-1, se toman los resultados de distancia de los 3 tipos de escenarios de cada modelo, así como la distancia única de 77.4m de la ruta preestablecida. Se obtienen las diferencias de distancia entre ruta preestablecida y los modelos propuestos Mom-AUV y Mol-AUV, además, la diferencia entre estos últimos. En el mismo orden con la ecuación 9, se obtienen los porcentajes de diferencia entre ellos.

$$Porcentaje \ Diferencia = \left(\frac{Distancia \ Diferencia}{Ruta \ preestablecida}\right) * 100$$
(9)

Escenario de densidad baja	Distancia entre Prestablecida y Mom-AUV (m)	Distancia entre Prestablecida y Mol-AUV (m)	Distancia entre Mom-AUV y Mol- AUV (m)	Porcentaje de Mom-AUV respecto a Preestablecida	Porcentaje de Mol-AUV respecto a Preestablecida	Porcentaje de Mol-AUV respecto a Mom-AUV
6	13.13	50.37	37.24	16.96	65.08	57.94

Promedio	18.50	57.35	38.85	23.90	74.09	66.28
42	13.02	42.84	29.82	16.82	55.35	46.32
36	22.21	64.00	41.79	28.70	82.69	75.72
33	13.36	54.71	41.35	17.26	70.68	64.57
27	17.02	59.25	42.23	21.99	76.55	69.94
15	22.18	55.73	33.55	28.66	72.00	60.76
14	17.80	60.83	43.03	23.00	78.59	72.20
13	18.46	61.09	42.63	23.85	78.93	72.33
12	20.60	62.03	41.43	26.61	80.14	72.94
11	22.28	63.77	41.49	28.79	82.39	75.27
10	20.01	58.66	38.65	25.85	75.79	67.35
9	23.24	56.83	33.59	30.03	73.42	62.02
8	21.90	62.76	40.86	28.29	81.09	73.62
7	13.82	50.02	36.20	17.86	64.63	56.94

En la Tabla 3-2, se muestran los resultados de las diferencias de porcentaje y distancia de los escenarios de baja densidad de estructuras. De esta, se puede resumir que Mom-AUV tiene un promedio de distancias menores entre la ruta preestablecida de 18.50m, representando un porcentaje del 23.90% menos distancia que la ruta preestablecida para estos escenarios. En cuanto a la rutas del modelo Mol-AUV con la prestablecida, se tiene un promedio de 57.35m de distancia menor, siendo un 74.09% menor que la preestablecida. Por otro lado, al comparar las rutas Mol-AUV se tiene un promedio de 38.85m de distancia de recorrido menores a Mom-AUV, el cual representa un 66.28% de ahorro en distancia.

Escenario de densidad media	Distancia entre Prestablecida y Mom-AUV (m)	Distancia entre Prestablecida y Mol-AUV (m)	Distancia entre Mom-AUV y Mol- AUV (m)	Porcentaje de Mom-AUV respecto a Preestablecida	Porcentaje de Mol-AUV respecto a Preestablecida	Porcentaje de Mol-AUV respecto a Mom-AUV
3	13.36	29.46	16.10	17.26	38.06	25.14
4	18.83	38.28	19.45	24.33	49.46	33.21
5	16.75	33.84	17.09	21.64	43.72	28.18
16	19.78	56.72	36.94	25.56	73.28	64.11
17	20.25	50.76	30.51	26.16	65.58	53.39
18	21.38	47.19	25.81	27.62	60.97	46.07
22	20.29	40.93	20.64	26.21	52.88	36.14
23	21.04	48.40	27.36	27.18	62.53	48.55
24	18.05	32.41	14.36	23.32	41.87	24.20
25	10.59	33.18	22.59	13.68	42.87	33.81
26	21.77	42.54	20.77	28.13	54.96	37.34
34	17.32	37.28	19.96	22.38	48.17	33.22
35	22.43	41.29	18.86	28.98	53.35	34.31
37	16.28	39.33	23.05	21.03	50.81	37.71

Tabla 3-3. Distancias y porcentajes de diferencia entre las rutas de los escenarios de densidad media.

Promedio	18.44	40.83	22.39	23.82	52.75	38.24

Para los escenarios de densidad media en la Tabla 3-3, las ventajas de distancia de Mom-AUV sobre la ruta preestablecida es de 18.44m en promedio, es decir, un 23.82% de distancia menor. Y para Mol-AUV es de 40.83m, lo que indica un 52.75% de ahorro en distancia. En relación a los dos modelos propuestos, Mol-AUV representa una ventaja de 22.39m en promedio, lo que es un 38.24% de recorrido menor que Mom-AUV.

Escenario de densidad alta	Distancia entre Prestablecida y Mom-AUV (m)	Distancia entre Prestablecida y Mol-AUV (m)	Distancia entre Mom-AUV y Mol- AUV (m)	Porcentaje de Mom-AUV respecto a Preestablecida	Porcentaje de Mol-AUV respecto a Preestablecida	Porcentaje de Mol-AUV respecto a Mom-AUV
1	11.11	13.09	1.98	14.35	16.91	2.99
2	8.47	23.98	15.51	10.94	30.98	22.50
19	16.94	48.46	31.52	21.89	62.61	52.13
20	11.64	20.14	8.50	15.04	26.02	12.93
21	19.09	28.20	9.11	24.66	36.43	15.62
28	11.05	16.79	5.74	14.28	21.69	8.65
29	14.63	27.85	13.22	18.90	35.98	21.06
30	13.32	32.28	18.96	17.21	41.71	29.59
31	10.57	13.29	2.72	13.66	17.17	4.07
32	14.74	25.08	10.34	19.04	32.40	16.50
38	7.75	10.08	2.33	10.01	13.02	3.35
39	15.75	14.83	-0.92	20.35	19.16	-1.49
40	23.64	33.02	9.38	30.54	42.66	17.45
41	5.09	24.86	19.77	6.58	32.12	27.34
Promedio	13.13	23.71	10.58	16.96	30.63	16.62

Tabla 3-4. Distancias y porcentajes de diferencia entre las rutas de los escenarios de densidad alta.

Para los escenarios de alta densidad de estructuras (Tabla 3-4), se sigue teniendo un ahorro de los dos modelos sobre la ruta preestablecida. En cuanto a Mom-AUV, en promedio 13.13m de distancia menor, que es un 16.96%. Para Mol-AUV un 23.71m, es decir, un 30.63% menos distancia que la preestablecida. Comparando Mom-AUV con Mol-AUV, el segundo tiene una distancia menor promedio de recorrido de 10.58m, siendo un 16.62% de recorrido menor que el primero.

3.4. CONCLUSIONES DEL SISTEMA DE GENERACIÓN AUTÓNOMA

En este capítulo 3 se presentó la propuesta de generación autónoma de rutas como una estrategia diferente a las existentes en tareas de monitoreo por vehículos autónomos. Al ser una propuesta en su etapa inicial, es probable que aún falten conceptos o situaciones para

definirse. Sin embargo, hasta este punto, los resultados generados dan confianza para continuar desarrollando y complementando el proyecto principal de construcción de un AUV para monitoreo autónomo del pez león.

Existen distintos puntos para concluir y se pueden clasificar en: la capacidad del sistema para obtener la profundidad, el planteamiento de lugares de búsqueda, la capacidad de localizar puntos de interés, la forma de evaluar con sesgos y los resultados finales de distancias.

a) Capacidad del sistema para obtener la profundidad

El ambiente fue simulado con distintas características visuales, incluyendo iluminación, sombras, colores, distancias y tamaños de objetos en metros, neblina (sustituyendo el agua marina con sedimentos), la vista de los escenarios se logró ver de manera similar a la real, como en la Figura 3-6 en la cámara del simulador, sin embargo no se tuvo la oportunidad de visualizar con la neblina desde las cámaras virtuales o probablemente faltó configurar algún parámetro. Con lo anterior, es probable que en un ambiente real, las cámaras no visualicen de la misma manera los objetos en el fondo, depende la sedimentación del agua, la iluminación y posibles peces como obstáculo para obtener la imagen de profundidad. Es necesario se trabaje en otra herramienta para la obtención de profundidad del ambiente marino.

Con esta simulación y la función de OpenCV, fue posible detectar objetos de 30 cm de alto como mínimo, pastos densos y no muy densos, piedras o arrecifes pequeños. Se detecta también los contornos de las estructuras que tienen diferencia de altitud y las marca como un nivel negativo, no como escala de grises, esto permitió distinguir las estructuras de un pequeño montículo que no tienen una pronunciada diferencia de altitud en sus orillas. En este avance, no se usó la altura de los objetos para una detección de altura de los mismos. Sólo información en 2 dimensiones para el procesamiento posterior, incluso la obtención de los puntos de interés carecen de valor en la altura. En un siguiente avance será tomado en cuenta esta dimensión.

b) Capacidad para obtener puntos de interés

El procedimiento para obtener puntos de interés logra colocar puntos alrededor de estructuras mayores a 1 metro aproximadamente, si es más pequeña, depende de su posición al tocar una de las líneas de intersección, si no es así, puede tener un punto encima, debido a la operación morfológica de dilatación para objetos entre 30 y 90 cm de

diámetro (aproximadamente), si es más pequeño y no tiene intersección, no se pondrá un punto cerca, con excepción de un punto predefinido en el modelo Mom-AUV.

c) Solución óptima de trayectorias

No en todos los casos el sistema de solución mediante TSP, específicamente la librería usada Cplex de IBM, resolvió el problema de ruta de la manera más óptima. En algunos casos, se observan cruces de trayectoria o retornos de la trayectoria en puntos cercanos a los ya visitados, como en los escenarios 1 y 40 en la Figura 3-18 para los 3 modelos de líneas de Mol-AUV. Algunos de estos casos sólo se pierden entre 1 y 3 metros, a simple vista. Hacer el cálculo manualmente para comprobar la distancia para una solución óptima, es tardado, en otro avance de la propuesta se llevará a cabo.

d) Evaluación por sesgos

Este proceso de evaluación ayuda a ver qué tan eficiente puede ser el monitoreo con las rutas obtenidas, no es el más objetivo, debido a que se suponen zonas de sesgo en una imagen (2 dimensiones), sin embargo, en esta investigación nos ayudó a definir el mejor modelo de líneas a utilizar, mientras más líneas más intersecciones con las estructuras generando puntos a su alrededor (reduce los sesgos).

e) Resultados del ahorro en distancias

Finalmente, con los porcentajes obtenidos en las Tablas 2, 3 y 4, promediando los porcentajes de los 3 tipos de escenario para las comparaciones entre los modelos, se tiene una distancia de recorrido menores de 21.56% de Mom-AUV y de 52.49% de Mol-AUV sobre la ruta preestablecida. Además, la diferencia entre los modos de operación de monitoreo propuestos, Mol-AUV representa un 40.38% en distancia menor de recorrido que Mom-AUV.

CAPITULO 4

METODOLOGÍA DEL SISTEMA DE RECONOCIMIENTO DEL PEZ LEÓN

El sistema de visión propuesto debe ser capaz de diferenciar de forma precisa el pez león de otras especies de peces dentro de su propio hábitat y resolver el problema desafiante del alto grado de mimetismo que presenta con el fondo marino, incluyendo la dificultad que puede presentar el procesamiento con imágenes y video en continuo movimiento (cámaras no fijas), en otras palabras, ser eficiente en condiciones no controladas.

El momento en el que el sistema de reconocimiento debe activarse durante la implementación de monitoreo, en esta propuesta, es cuando el AUV esté desplazándose por el fondo marino, visitando cada punto de interés encontrados por el sistema de generación autónoma de trayectorias (Figura 4-1).





4.1. MODELOS PROBADOS PARA EL SISTEMA DE RECONOCIMIENTO

Es importante mencionar que antes de definir cuál sería el modelo o técnicas de reconocimiento de objetos para nuestro sistema, se realizaron una serie de pruebas no muy detalladas, con la finalidad de ver de una manera rápida la capacidad para reconocer el pez león.

4.1.1. Descriptor HOG y clasificador SVM

La primera arquitectura basada en los métodos tradicionales de ML, ampliamente usada y con excelentes resultados en una gran variedad de objetos y en condiciones semi-controladas [124, 125, 126, 127] es el descriptor HOG junto con el clasificador SVM. El proceso consistió en un

entrenamiento de 3 modelos; entrenamiento del cuerpo completo del pez león, el segundo sólo cuerpo sin tomar en cuenta las espinas y el tercer modelo sólo la cabeza del pez (Figura 4-2).



Figura 4-2. Representación de los gradientes orientados; a) Cuerpo y espinas para el modelo 1, b) Solo cuerpo para modelo 2 y c) Sólo cabeza para el modelo 3.

En resumen sobre esta arquitectura; los resultados fueron muy bajos para los tres modelos el mayor de 25.54% de precisión, incluso reuniendo los 3 modelos como uno solo (cuarto modelo) en la pruebas, su precisión fue de 36.46%. Por tal motivo, se continuó probando distintas opciones.

4.1.2. Redes Neuronales Convolucionales

El segundo modelo fue la implementación de las CNN perteneciente a DL. Por si sólo como método, funciona como un clasificador, es decir, puede entrenarse y dar como resultado la existencia del pez león o no, sin la detección en una determinada zona en la imagen. Por esta razón, fue adaptado un procedimiento con PDI con ayuda de la librería OpenCV, para convertirlo en un sistema de detección o reconocimiento del pez león.

La Figura 4-3, representa la configuración de la red neuronal con 4 etapas de convolución (generación de nuevas imágenes a partir de filtros) y maxpooling (operaciones para reducir el tamaño de las imágenes), finalmente, capas de clasificación para decidir si es el pez león o el fondo marino. Obviando más detalles de la configuración y la etapa posterior para crear el sistema de reconocimiento, solo es importante mencionar que se observó un buen desempeño para detectar el pez león en condiciones no controladas, no fue necesario obtener métricas para cuantificar el desempeño, debido al tiempo excesivo de 4 a 6 segundos de procesamiento para una sola imagen. Por lo anterior, se descartó este sistema.


Figura 4-3. Modelo implementado de Redes Neuronales Convolucionales (CNN) para el Pez León.

4.1.3. YOLOv2

El tercer modelo implementado fue YOLOv2 [128], incorpora CNN, tiene una versión ligera llamada "tiny", con una capacidad procesamiento hasta de 10 imágenes por segundo sin GPU. Al finalizar las pruebas rápidas, se obtuvieron resultados de hasta 76.46% de mAP.



Figura 4-4. Diagrama del sistema de detección del pez león con YOLOv2.

En la figura 4-4 se observa la metodología seguida en la implementación del modelo YOLOv2, con una base de datos de 644 imágenes que incluyeron 2407 peces león para la etapa de

entrenamiento. Este modelo resultó tener resultados muy buenos sin tener una gran base de datos.

Durante el proceso de investigación e implementación de estos modelos, se publicaron más algoritmos con una velocidad similar, incluso mayores [129, 130]. Además, en el presente proyecto se decidió utilizar la tarjeta de Nvidia Jetson Nano, por ser pequeña, bajo consumo energético y alta capacidad de cómputo (descrita a detalle más adelante). En consecuencia se buscaron las versiones más actuales de mayor desempeño y que fueran compatibles con la tarjeta de Nvidia. Aunque los métodos anteriores no hayan sido los adecuados para cumplir con la tarea, se considera importante mencionarlos para que se conozca cuales metodologías no son adecuadas para ello.

A continuación se presenta la metodología final del sistema de reconocimiento del pez león, comparando 2 metodologías de gran desempeño y velocidad, compatibles con sistemas embebidos como la tarjeta Jetson Nano.

4.2. IMPLEMENTACIÓN CON ALGORITMO YOLOv4 Y SSD-MobileNet-v2

El sistema de reconocimiento YOLOv4 [130] es una versión mejorada de la implementación expuesta anteriormente, tiene similares desempeños en cuanto a mAP (de 48.1% hasta un 66%) en el set de datos COCO [131], además, de mejorar la velocidad de procesamiento en dispositivos con GPUs. Inclusive, es posible la implementación de este sistema en la tarjeta embebida Jetson Nano de NVIDIA de bajo consumo energético, ahora usando TensorRT [132] en lugar de Tensorflow [133]. De similar desempeño el sistema SSD-MobileNet [134] también fue implementado en la misma tarjeta, basándonos en la implementación SSD-MobileNet-v2 [135] sobre el framework Pythorch [136]. Esta vez, ambos sistemas se someten a un riguroso y confiable proceso de validación; a una validación cruzada de 5 veces. En la Figura 4-5 se observa la metodología en bloques que seguirán ambos sistemas.



Figura 4-5. Diagrama de los sistemas de detección de YOLOv4 y SSD-MobileNet-v2 con Validación Cruzada de 5k.

La metodología de este sistema de reconocimiento comienza con la adquisición de un conjunto de imágenes del fondo marino que contiene el pez león, este conjunto de datos se separa luego de pasar la etapa de etiquetado en datos de entrenamiento y en datos de prueba, al realizar el entrenamiento con Validación Cruzada de 5 veces, se obtienen cinco modelos entrenados, los cuales son evaluados con el conjunto de prueba y, finalmente, el mejor modelo es evaluado a detalle con los tres subconjuntos de prueba con diferente iluminación. (Figura 4-5). El sistema de detección probado en este trabajo es YOLO v4, con la configuración Tiny usando TensorRT y el segundo sistema es SSD-Mobilenet-v2 que usa Pytorch. Una de las ventajas de estos dos sistemas, además del desempeño mostrado, es que recientemente fueron liberados y probados por el equipo de NVIDIA en la tarjeta Jetson Nano. Estos informes están disponibles en [137].

4.2.1. Equipo utilizado

Hardware

- Jetson Nano Nvidia: 4GB de RAM, Procesador Quad-core ARM A57 a 1.43 GHz y un GPU integrado Tegra X1 149-core.
- 2 memorias SD 64GB, para montar el sistema operativo y configurar sistemas de reconocimiento.
- Pantalla LCD.
- Cables para conexión.

Software

- Nvidia JetPack SDK 4.4: contiene Ubuntu Linux 18.04 LTS.
- Sistema YOLOv4 [138].

• Sistema SSD-MobileNet-v2 [139].

4.2.2. Base de datos

A diferencia de la implementación anterior de YOLOv2, para estos dos sistemas se aumenta la base de datos a un total de 2818 imágenes que incluyen 6466 peces león; 1578 imágenes las obtuvo nuestro equipo de investigación del UCIA en áreas arrecifales de Cancún, Puerto Morelos y Tulum, de estas, 1489 imágenes se obtuvieron de 10 grabaciones en video. Las restantes 1240 imágenes se obtuvieron de internet (Google, YouTube y NaturaLista), de las cuales 903 fueron obtenidas de 16 videos de YouTube.

La base de datos fue dividida en dos conjuntos; el primero contiene 2360 imágenes correspondientes al proceso de entrenamiento con validación cruzada de 5 veces. Este conjunto es dividido a su vez, alternadamente (en las 5 veces que se entrena), en 480 imágenes (20%) de validación y 1880 imágenes de entrenamiento (80%). Mientras que el segundo conjunto restante de 458 imágenes son asignadas al conjunto de prueba.

El conjunto prueba, a su vez, es dividido en tres subconjuntos para evaluar la capacidad del sistema en detectar el pez león en 3 tipos de iluminación (baja, media y alta). En la Figura 4-6 se muestran los 3 tipos de iluminación de los subconjuntos de prueba.



Figura 4-6. Subconjuntos de prueba en relación a la iluminación presente en el pez; a) iluminación baja, b) media y c) alta.

4.2.3. Etiquetado de imágenes

Ambos sistemas de detección, además de la transferencia de aprendizaje, utilizan una base de datos para aprender. Para etiquetar todo el conjunto de datos se utilizó la herramienta labellmg [140] de cada imagen etiquetada se obtuvo un archivo *.xml* con el nombre de etiqueta "PezLeon" y suscoordenadas (x, y) de la posición de los peces león.

4.2.4. Implementación

Para YOLOv4 se utilizó la versión "tiny" y transferencia de aprendizaje con un modelo preentrenado en el conjunto de datos VOC 2007-2012, adicionalmente se configuró es sistema con los hiperparámetros recomendados adaptándo sólo el número de clases a 1; un tamaño de lote de 32, un Lr de 0.00261, y un número de iteraciones de 6000, que corresponde a 100 épocas. Además, durante el proceso de entrenamiento, se realizó la validación cruzada 5 veces.

En el caso de SSD-MobileNet-v2, también tiene transferencia de aprendizaje con el conjunto de datos de VOC, se configuro bajo las recomendaciones del sistema a una clase, un tamaño de lote de 2 (debido a la capacidad de procesamiento de la Jetson Nano), un Lr de 0.001 y 100 épocas.

4.2.5. Procedimiento de validación cruzada

Esta metodología nos permite validar los resultados de un sistema y comparar el desempeño entre varios sistemas y evitar un sobre-ajuste. Es un procedimiento que se realiza durante el entrenamiento de la siguiente manera:

- La cantidad de imágenes o base de datos de entrenamiento (2360) se subdivide en aproximadamente 80% para entrenamiento (1880 imágenes) y 20% para validación (480 imágenes).
- Se configura el sistema para que después de ciertas épocas de entrenamiento realice una evaluación automática en las imágenes de validación. Comienza el entrenamiento (primera vez).
- Nuevamente se subdivide 80% y 20% entrenamiento y validación pero con imágenes distintas a la anterior validación, es decir, por cada vez que se entrene (5 veces) los datos de entrenamiento deben rotarse en los datos de validación.

 Se realiza 5 veces el entrenamiento por cada sistema, obteniendo en cada uno un modelo entrenado. Los modelos obtenidos por cada sistema también pueden evaluar los datos de prueba

Algunas métricas para evaluar el desempeño son aplicadas automáticamente cuando se realiza la validación, durante el entrenamiento. Después de este procedimiento, con el modelo de cada entrenamiento obtenido, se evalúa el desempeño con los datos de prueba de uno o varios sistemas, estos datos son imágenes no usadas para entrenamiento ni validación.

4.2.6. Métricas para evaluar el desempeño.

Para validar el rendimiento del modelo entrenado en nuevas imágenes (prueba) en sistemas de detección de objetos, se utilizan las siguientes métricas.

• Intersección sobre unión (IoU)

También conocido como el índice Jacard. Es una métrica que permite determinar el grado de precisión con el que el modelo entrenado puede detectar el objeto. La Figura 4-7 muestra cómo se calcula gráficamente. Los marcos marcados manualmente se comparan al etiquetar con los marcos que resultaron de la predicción, si hubiera más predicciones cercanas, se toma la más cercana y se cancela para continuar la búsqueda de coincidencias, para evitar múltiples detecciones del mismo objeto.





• Sensibilidad (recall)

Determine qué tan bueno es el modelo para detectar objetos de clase de interés. Es decir, cuántos peces son verdaderos positivos (TP, por True positives) del total de peces en los datos de prueba, que es la suma de TP y los falsos negativos (FN, por False Negative).

$$Recall = \frac{TP}{TP + FN}$$
(37)

Precisión

Evalúa la habilidad para tener resultados correctos del total de predicciones.

$$Precisión = \frac{TP}{TP + FP}$$
(38)

Donde FP es Falsos Positivos.

• Media del promedio de precisiones (mAP)

Es el promedio de las precisiones máximas a diferentes valores de sensibilidad en diferentes clases a detectar. El modelo asigna estos valores de precisión cuando se realiza la predicción. En este caso, solo hay una clase, por lo que también podemos llamarla Precisión Promedio (AP) del modelo.

4.3. RESULTADOS Y DISCUSIONES DEL CAPÍTULO

El análisis de los resultados en este apartado nos permite justificar la elección de uno de los dos sistemas de reconocimiento propuestos en este trabajo. Además de examinar las métricas de rendimiento de este sistema, estas se compararán con algunos de los sistemas de reconocimiento del pez león mencionados en el estado del arte.

4.3.1. Resultados del sistema de reconocimiento del pez león

Los resultados de la etapa de entrenamiento con validación cruzada para los sistemas YOLOv4 y SSD-MobileNet-v2 se presentan en la Tabla 4-1. En el primer sistema hay una media de 78,22% mAP para los mejores desempeños en la validación y una media de 1,44 para el promedio de pérdida de entrenamiento. El mAP promedio es 66.21% para la evaluación del conjunto de prueba obtenido de los 5 modelos con un valor de loU de 0.50.

Para el segundo sistema SSD-Mobilenet-v2, se obtiene un mAP promedio de 63.84% de los 5 modelos evaluados con Cross Validation, un promedio de 2.91 de las pérdidas de entrenamiento promedio, y un mAP promedio de 57.49% (con IoU = 0.50) para la evaluación del conjunto de prueba.

Sistema	No. de validación cruzada	Mejor mAP % de validación	Mejor promedio de pérdida en entrenamiento	mAP en resultados de prueba (loU=0.5)	
YOLOv4 tiny	1	76.47	1.2	62.78	
	2	91.09	1.4	63.57	
	3	69.02	1.7	70.60	
	4	81.22	1.5	62.03	
	5	73.33	1.4	72.05	
Promedio		78.22	1.44	66.21	
SSD- MobileNet-v2	1	58.95	3.51	62.88	
	2	79.43	2.27	63.48	
	3	58.69	2.83	63.94	
	4	51.71	3.2	35.92	
	5	70.46	2.76	61.25	
Promedio		63.84	2.91	57.49	

Tabla 4-1. Resultados de validación cruzada de 5-veces en el entrenamiento con los sistemas YOLOv4 y SSD-MobileNet-v2.

El modelo mejor evaluado con el proceso de validación cruzada es el número 5 del sistema YOLOv4, que tiene un valor de mAP más alto 72.05% (con IoU = 0.50) del conjunto de prueba. Además, se realizó una evaluación de los tres subconjuntos de prueba con iluminación baja, media y alta con diferentes valores de umbral de IoU de 0,75 a 0,05 con pasos de 0,05 (la Tabla 4-2 muestra solo resultados para 0,25 y 0,50).

Métricas	Prueba general		lluminación baja		Iluminación media		lluminación alta	
IoU	0.25	0.50	0.25	0.50	0.25	0.50	0.25	0.50
TP	466	445	187	180	159	152	120	113
FP	53	74	15	22	22	29	16	23
FN	266	287	120	127	76	83	72	79
mAP %	79.15	72.05	78.40	71.41	78.18	71.86	80.72	72.22
Precisión %	89.79	85.74	92.57	89.1	87.85	83.98	88.24	83.09
Sensibilidad								
(Recall) %	63.66	60.79	60.91	58.63	67.66	64.68	62.50	58.85

Tabla 4-2. Resultados de las métricas de evaluación del conjunto y subconjunto de pruebas con el modelo YOLOv4.

En negrita, los valores más altos que corresponden al valor de IoU =0.25 para cada subconjunto.

El sistema YOLOv4 demostró durante las evaluaciones de validación y pruebas un tiempo de procesamiento, de 480 imágenes por 33.4 segundos en promedio, lo que resulta en 14.37 imágenes por segundo. En el anexo 5 se muestran capturas de pantalla de los resultados para los 5 modelos obtenidos en la validación cruzada para el sistema YOLOv4.

Algunos de los casos de FP obtenidos en la evaluación del equipo de prueba se muestran en la Figura 4-8; hay una detección de TP en cada subimagen; así como detecciones de 2 FN en las subimágenes (a), (b) y (c); y un caso de detección de FN en (d). Por otro lado, estas imágenes tienen un fondo marino semiduro (semi-rocoso) lo que representa un desafío para los sistemas de detección, debido a que las rocas y la vegetación marina pueden generar siluetas similares al pez león.



Figura 4-8. Resultados de detección de FP y TP; 4 peces león identificados correctamente (TP) y 7 falsas identificaciones.

Hay 266 FN en la evaluación de la prueba, que representan los casos en los que el sistema no detecta el pez león, como en los 12 casos en total de las subimágenes (a), (b) y (c) o no detectado adecuadamente como en la subimagen (d) de la Figura 4-9. Es posible observar la

variedad de imágenes en el conjunto de prueba, como la resolución, la nitidez debido al agua, los sedimentos y / o la distancia, la iluminación y la oclusión de los peces como en (b) y (d).



Figura 4-9. Resultados de detección de FN y TP; 7 peces león correctamente identificados (TP) y 12 faltantes para identificar (FN), incluido el pez león de la subimagen (d).

La Figura 4-10 muestra múltiples detecciones correctas de pez león con más del 67% de confianza. En la subimagen 6(b) se puede observar un pez león que causa una pequeña oclusión en tres peces león en la parte superior derecha. Además, se pueden ver pequeñas superposiciones en los cuadros de detección incluso con una identificación correcta.

Otro aspecto a considerar, además de que el complejo fondo marino puede confundirse con el pez león, es que el pez león puede confundirse con el fondo marino. En la Figura 4-11 se presentan dos ejemplos de mimetismo.





Figura 4-10. Múltiples identificaciones del pez león con altos valores de loU y de confianza.



Figura 4-11. Resultados de alto grado de mimetismo causados por la forma y el color del fondo marino; el sistema sólo identificó correctamente al pez en 7(b)

En la Figura 4-12 Se presentan dos imágenes de las cámaras de 180 ° que obtuvo el equipo de investigación, donde se pueden observar dos casos de baja iluminación detectados correctamente.



Figura 4-12. Detección con baja iluminación en imágenes de 180°

Se muestran en el anexo 6 más ejemplos de resultados de imágenes analizadas con el sistema YOLOv4.

4.3.2. Análisis y discusiones

El promedio de resultados de mAP de los cinco modelos en la validación cruzada para el sistema YOLOv4 fue de 14.34% sobre el SSD-MobileNet-v2 (Tabla 4-1). Además, en la evaluación de los datos de retención de los cinco modelos, se observó una ventaja del sistema YOLOv4 con un 8,72% en el mAP promedio (cuando el valor de loU es 0,50). Además, el mejor valor de 72,05% mAP se encontró en el modelo número cinco. De lo anterior, se muestra que el sistema YOLOv4 tuvo un mejor desempeño para la detección de pez león que SDD-MobileNet-v2.

En un análisis más detallado, YOLOv4 con una variación en IoU (Tabla 4-2) tiene un aumento en mAP de 7.1%, 4.04% en precisión y 2.87% en recuerdo. Lo anterior se obtiene con un valor de IoU = 0,25 en lugar del valor predefinido de IoU = 0,50. Esto significa que el umbral, que define si las identificaciones son positivas o no, se amplía y, por lo tanto, los cuadros de identificación de detección no necesitan ser precisos con los cuadros de etiquetado para ser considerados verdaderos positivos.

En el subconjunto de prueba con las tres iluminaciones (Figura 4-6), se pueden distinguir diferentes características en el pez león. Por ejemplo, en la iluminación alta, las rayas y las

espinas se observan claramente, mientras que en la iluminación baja, las rayas ya no son perceptibles y, en algunos casos, solo se ven las espinas; por lo tanto, la forma del cuerpo del pez ya no es perceptible. La Tabla 4-2 muestra que un valor de IoU = 0.25 tiene el mejor desempeño de precisión (92.57%) en baja iluminación, aproximadamente un 5% más que los demás; esto significa que tiene más detecciones positivas que el número total de detecciones realizadas por el sistema para este subconjunto. En el caso de iluminación media, hay menos casos de detección en los que el pez león no está presente; este comportamiento se deduce del valor de recuperación ligeramente superior. El desempeño de los subconjuntos de prueba del sistema de detección no presentó diferencias notables en términos de valores de mAP para los tres tipos de iluminación, a pesar de tener diferentes cantidades de imágenes y peces (Tabla 4-2); esto significa que el sistema es estable en la detección de las diferentes características presentes en el pez león en su entorno natural.

En cuanto a las métricas obtenidas de la evaluación del conjunto de prueba (Tabla 4-2), se obtiene una precisión del 89,79% y una recuperación del 63,66%. Sin embargo, no es posible tener una comparación simple con el trabajo de Naddaf y su equipo [57], quienes evalúan su sistema en un videoclip presentando una precisión del 91% con una confianza en la detección del 85%. Primero, no presentan el valor de configuración de IoU, lo que no permite observar una evaluación precisa para definir TP, FP y FN; En las figuras 6, 12, 13, 15 y 18 del artículo mencionado se observan ejemplos de TP en los que podría asumirse un valor umbral de IoU por debajo de 0,25 en los recuadros de detección. En segundo lugar, los resultados presentados se basan en el análisis de un video, que no permite evaluar el rendimiento de su sistema para diferentes posiciones, fondos y presentaciones de iluminación sobre el pez león en su entorno natural. Por lo anterior, no se puede hacer una comparación más clara con los resultados de este trabajo.

En la Figura 4-8, donde se muestran las detecciones de FP, también se observan ciertos elementos como peces, corales, algas, rocas y otros organismos. Esto está relacionado con la posición e iluminación de la captura de imagen, se generan formas similares al pez león, lo que hace un escenario complejo para el sistema de detección. Incluso con esta limitación del entorno, el número de 53 FP obtenidos en la evaluación de la prueba representan el 10,21% del total de predicciones realizadas por el sistema, lo que da un buen porcentaje complementario a la precisión del sistema (89,79%). Sin embargo, los casos de falsos negativos son elevados, estos son los casos de pez león no detectados que representan el 36,34%. Por ejemplo, en la Figura 4-9a, hay ocho casos de falta de identificación, probablemente debido a la baja

resolución de la imagen, la turbidez del agua o la poca iluminación. Un caso similar a los 3 FN es en la Figura 4-9c donde se suma el factor de distancia de estos tres peces. Las Figuras 4-9b y 4-9c muestran imágenes más nítidas, sin embargo, en el caso de la primera, el pez está ocluido y las espinas no son muy visibles. En la Figura 4-9d hay una detección que cubre solo parcialmente al pez león; en este caso, el umbral definido por IoU entre el cuadro de detección y el cuadro de etiqueta está por debajo del valor definido 0,25.

4.4. CONCLUSIONES DEL SISTEMA DE RECONOCIMIENTO

En este trabajo, hemos comparado dos sistemas de reconocimiento de objetos recientes y destacados; elegimos el sistema YOLOv4 por tener el mejor rendimiento en la detección del pez león.

Este sistema de detección de pez león presenta un análisis más completo que en el artículo de Naddaf y su equipo [55] y el informe de Antaya y colaboradores [57]. En el primero, falta un análisis de IoU y en consecuencia de mAP. En el segundo informe, tienen su mejor rendimiento mAP del 31,5% para su modelo SSDlite-MobileNet-v2. Por otro lado, ahora hemos obtenido un mAP del 79,15%, una precisión del 89,79% y un valor de recuperación del 63,66%, con un valor de umbral de IoU de 0,25.

El análisis del subconjunto de prueba relacionado con la iluminación presentada en el pez león mostró que el sistema es estable en cualquiera de los casos evaluados; esto significa que nuestro sistema propuesto es estable en las condiciones del entorno incontrolado del pez león.

En cuanto a los resultados del sistema de reconocimiento en las tres diferentes condiciones de iluminación, podemos concluir que el sistema basado en CNNs ha demostrado ser muy eficiente en el aprendizaje de patrones complejos, es versátil y robusto en un entorno no controlado donde se encuentra el pez león, incluso si hay poca iluminación con variados tipos de fondo y diferentes perfiles o posturas del pez león. Para seguir trabajando, agregaremos a nuestra base de datos más ejemplos de mimetismo para mejorar la robustez de nuestro modelo.

Existen pocos casos de evaluación de imágenes con mimetismo. Sería beneficioso evaluar más a fondo el comportamiento del sistema en estos casos. También se necesitan más conjuntos de datos con mimetismo para entrenar y evaluar.

Finalmente, se observó un buen desempeño en múltiples detecciones (Figuras 4-10) y también en lugares con poca iluminación (Figura 4-12), especialmente cuando la calidad de imagen es alta.

CONCLUSIONES GENERALES

Aun es complicado saber o clasificar el nivel actual de autonomía que pueden tener los AUV, no es comparable con los niveles de autonomía en vehículos sobre tierra definidos y publicados en 2018 por SAE internacional (Society of Automotive Engineering) [141], debido al ambiente en el que se desenvuelven y las misiones que estén destinadas a hacer, sin embargo, sí se puede asegurar que se tendrá mayor autonomía en vehículos submarinos cuando hagan más uso de la inteligencia artificial para la navegación mediante el reconocimiento del fondo marino, la fusión de datos de distintos sensores, identificación de blancos, detección de errores y la autoreconfiguración, y así, limitar la participación del ser humano en un mínimo de intercambio de información.

El diseño de este sistema de monitoreo está basado en el comportamiento del ser humano cuando busca algo en un sitio; visualiza el espacio, decide los sitios probables de búsqueda, asigna prioridades para ordenar y hacer eficiente la tarea, finalmente, se busca el objeto en los lugares seleccionados en el orden previsto. Si logra realizarse esto dentro de un vehículo autónomo, además del seguimiento de trayectorias, obtención de energía y almacenamiento, la comunicación inalámbrica y la capacidad de procesamiento que lo hacen un vehículo autónomo, se estaría aportando más autonomía con este modelo de monitoreo. Este tipo de monitoreo es enfocado a especies u objetos, de los cuales se debe conocer los lugares probables o su comportamiento, para diseñar previamente la localización automática de los puntos de interés.

Esta propuesta de generación autónoma de trayectorias para el monitoreo del pez león, además de dar aumento en la autonomía de un AUV, reduce hasta un 54.27% las distancias de recorrido de las distancias generadas por la ruta preestablecida que actualmente es utilizada para aplicaciones de AUV en tareas de monitoreo de especies. Esto porcentaje de ahorro se puede representar también en un ahorro de energía que puede consumir el AUV durante su trayectoria.

Al cambiar la situación con la pandemia de la enfermedad Covid-19 en todo el mundo, cambió también la manera de continuar con este trabajo de investigación. El ambiente simulado en 3D

en Gazebo y ROS con ayuda de Blender, han sido una herramienta fundamental para la realización de las pruebas, tomando en cuenta que no estaba planeado de este modo, se ha logrado completar los objetivos planteados en un inicio. Además, aún se puede seguir aprendiendo de las herramientas para hacer uso de su potencial como las simulaciones con movimiento y el uso de diversos sensores virtuales, para así, evaluar y demostrar propuestas como esta.

En cuanto al sistema de reconocimiento del pez león, se encontraron algunas áreas de oportunidad para mejorar la implementación de nuestro algoritmo, son parte de los desafíos que se presentan donde no se controlan las condiciones de la configuración óptica y el objeto a reconocer; debemos contar con alternativas como el uso de cámaras con mayor resolución, una lámpara auxiliar, así como acortar distancias cuando el AUV busque de forma autónoma el pez león. De esta forma podríamos reducir la falta de identificación (FN) (que representa el 36,34% en nuestros resultados) para mitigar el alto grado de mimetismo al tiempo que reducimos el número de identificaciones falsas que representa el 10,21%.

Paralelamente a este trabajo, se han logrado avances importantes en investigaciones que forman parte del proyecto, como en el diseño y control de trayectorias de un vehículo marino de superficie [86], el diseño del AUV tipo torpedo con cuatro grados de libertad [142], un sistema de gestión de energía para una microrred de CC (panel fotovoltaico y baterías) resuelto mediante algoritmos genéticos y enjambre de partículas [143], así como un convertidor elevador-reductor para la regulación de tensión de salida [144].

Este trabajo representa un avance modular en el diseño del proyecto que busca realizar tareas de monitoreo del pez león a través de un AUV; mediante el reconocimiento del pez león y reduciendo el consumo de energía a través de la disminución de distancias de las trayectorias en comparación con la forma convencional de trayectorias predeterminadas utilizadas actualmente por vehículos autónomos. En este sentido, hemos publicado un trabajo en relación al reconocimiento del pez león [145] y trabajamos actualmente en el relacionado a la generación autónoma de trayectorias.

TRABAJO FUTURO

Del sistema de generación de trayectorias

- Agregar la altitud de los puntos de interés y la perturbación de la corriente en nuevas evaluaciones.
- Agregar movimiento al AUV simulado en Gazebo-ROS
- Se puede hacer una evaluación de los sesgos más completa en el espacio virtual de Gazebo-ROS, con la posibilidad de implementar SLAM.
- Incluir un segundo vehículo Autónomo (vehículo de superficie) para el mapeo.
- Probar la viabilidad del sistema de mapeo con visión estéreo en fondo marino real.
- Probar distintos sistemas de mapeo marino; si se aumenta la zona de mapeo podría aumentarse el área de monitoreo en cada inmersión o búsqueda.
- Probar la programación de algún método distinto de solución de ruta óptima.

Del sistema de reconocimiento del pez león

- Seguir aumentando la base de datos del pez león para reentrenar el sistema y mejorar los resultados de las métricas.
- Probar el sistema de reconocimiento con la cámara virtual de resolución alta y la cámara real de alta resolución en el fondo marino real, así como, su velocidad de procesamiento.
- Probar el sistema de reconocimiento del pez león en Gazebo-ROS con movimiento del AUV, cámara virtual y peces león simulados.

BIBLIOGRAFÍA

[1] Morris, J. A., Akins, J. L. (2009) Feeding ecology of invasive lionfish (Pterois volitans) in the Bahamian archipelago. Environmental Biology of Fishes. 86(3), 389-398 https://doi.org/10.1007/s10641-009-9538-8

[2] Madrid, J., Sánchez, P. Ruiz, A. (1997) Diversity and Abundance of a Tropical Fishery on the Pacific Shelf of Michoacán, México. Estuarine, Coastal and Shelf Science. 45(4), 485–495 <u>https://doi.org/10.1006/ecss.1996.0219</u>

[3] Murphy, H. M., Jenkins, G. P. (2010) Observational methods used in marine spatial monitoring of fishes and associated habitats: a review. Marine and Freshwater Research, 61(2), 236-252

https://doi.org/10.1071/MF09068

[4] Edgington, D. R., Cline, D. E., Davis, D., Kerkez, I., Mariette, J. (2006) Detecting, tracking and classifying animals in underwater video. En: IEEE OCEANS 2006. 18-21 Septiembre. Boston, MA, USA.

[5] Schofield, P. J. (2009) Geographic extent and chronology of the invasion of non-native lionfish (Pterois volitans [Linnaeus 1758] and P. miles [Bennett 1828]) in the Western North Atlantic and Caribbean Sea, Aquatic Invasions. 4(3), 473-479

[6] Courtenay, W.R. (1995) Marine fish introductions in southeastern Florida, American Fisheries Society Introduced Fish Newsletter. 14(1), 2–3

[7] Hamner, R. M., Freshwater, D. W., Whitfield, P. E. (2007). Mitochondrial cytochrome b analysis reveals two invasive lionfish species with strong founder effects in the western Atlantic. Journal of Fish Biology. 71, 214-222

https://doi.org/10.1111/j.1095-8649.2007.01575.x

[8] Arredondo-Chavez, A. T., Sanchez-Jimenez, J. A., Avila-Morales, O. G., Torres-Chavez, P., Herrerias-Diego, I., Medina-Nava, M., Madrigal-Guridi, X., Campos-Mendoza, A., Dominguez-Dominguez, O., Caballero-Vazquez, J. A. (2016) Spatio-temporal variation in the diet composition of red lionfish, Pterois volitans (Actinopterygii: Scorpaeniformes: Scorpaenidae), in the Mexican Caribbean: insights into the ecological effect of the alien invasion. Acta Ichthyologica et Piscatoria 46(3):185–200.

https://doi.org/10.3750/aip2016.46.3.03

[9] Albins, M. A., y Hixon, M. A. (2008) Invasive Indo-Pacific lionfish *Pterois volitans* reduce recruitment of Atlantic coral-reef fishes, Marine Ecology Progress Series. 367, 233-238 https://doi.org/10.3354/meps07620

[10] Albins, M. A., Hixon, M. A. (2013) Worst case scenario: potential long-term effects of invasive predatory lionfish (*Pterois volitans*) on Atlantic and Caribbean coral-reef communities, Environmental Biology of Fishes. 96(10-11), 1151-1157 https://doi.org/10.1007/s10641-011-9795-1

[11] Bernal, N. A., DeAngelis, D. L., Schofield, P. J., Sealey, K. S. (2015) Predicting spatial and temporal distribution of Indo-Pacific lionfish (Pterois volitans) in Biscayne Bay through habitat suitability modeling, Biological Invasions. 17(6), 1603-1614 https://doi.org/10.1007/s10530-014-0819-6

[12] Semmens, B. X., Buhle, E., Salomon, A., Pattengill-Semmens, C. (2004) A hotspot of nonnative marine fishes: evidence for the aquarium trade as an invasion pathway, Mar Ecol Prog Ser. 266, 239–244

[13] Morris, J. A., Jr. (2013) El pez león invasor: guía para su control y manejo. Gulf and Caribbean Fisheries Institute Special Publication Series, No. 2, Marathon, Florida, USA. pp 126.

[14] Burger, W., Burge, M. (2015) Digital Image Processing. An algorithmic Introduction Using Java. Second Edition. Springer, London.

[15] Szeliski, R. (2010) Computer vision: algorithms and applications. Springer Science & Business Media, New York.

[16] Arbour, J. H., Hardie, D. C., Hutchings, J. A. (2011) Morphometric and genetic analyses of two sympatric morphs of Arctic char (Salvelinus alpinus) in the Canadian High Arctic. Canadian Journal of Zoology, 89(1), 19-30.

https://doi.org/10.1139/Z10-100

[17] Spampinato, C., Giordano, D., Di Salvo, R., Chen-Burger, Y. H. J., Fisher, R. B., Nadarajan, G. (2010) Automatic fish classification for underwater species behavior understanding. En: Proceedings of the first ACM international workshop on Analysis and retrieval of tracked events and motion in imagery streams, 29 Octubre. Firenze, Italia.

https://doi.org/10.1145/1877868.1877881

[18] Hsiao, Y. H., Chen, C. C., Lin, S. I., Lin, F. P. (2014) Real-world underwater fish recognition and identification, using sparse representation, Ecological informatics. 23, 13-21 https://doi.org/10.1016/j.ecoinf.2013.10.002

[19] Haimovici, M., Cardoso, L. G., Unpierre, R. G. (2016) Stocks and management units of Micropogonias furnieri (Desmarest, 1823) in southwestern Atlantic, Latin American Journal of Aquatic Research. 44(5), 1080-1095 https://doi.org/10.3856/vol44-issue5-fulltext-18

[20] Heithaus, M. R. Dill, L. M. (2002) Food availability and tiger shark predation risk influence bottlenose dolphin habitat use, Ecology. 83(2), 480-491 https://doi.org/10.1890/0012-9658(2002)083[0480:FAATSP]2.0.CO;2

[21] Costa, C., Antonucci, F., Boglione, C., Menesatti, P., Vandeputte, M., Chatain, B. (2013) Automated sorting for size, sex and skeletal anomalies of cultured seabass using external shape analysis. Aquacultural engineering, 52, 58-64 https://doi.org/10.1016/j.aquaeng.2012.09.001

[22] Ferrero, F. J., Campo, J. C., Valledor, M., Hernando, M. (2014) Optical systems for the detection and recognition of fish in rivers. En: IEEE 11th International Multi-Conference on Systems, Signals & Devices (SSD14). 11-14 Febrero. Barcelona, España.

[23] Beyan, C., Fisher, R. B. (2012) A filtering mechanism for normal fish trajectories. En: IEEE Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012). 11-15 Noviembre. Tsukuba, Japón.

[24] Zhang, D., Lee, D. J., Zhang, M., Tippetts, B. J., Lillywhite, K. D. (2016) Object recognition algorithm for the automatic identification and removal of invasive fish. Biosystems Engineering, 145, 65-75.

https://doi.org/10.1016/j.biosystemseng.2016.02.013

[25] Wang, Y., Ye, H. Li, B. (2010) A research based on recognition algorithm of characteristics of body surface of infected fish. En: IEEE 2010 World Automation Congress. 19-23 Septiembre. Kobe, Japón.

[26] Hu, J., Li, D., Duan, Q., Han, Y., Chen, G., Si, X. (2012) Fish species classification by color, texture and multi-class support vector machine using computer vision. Computers and electronics in agriculture. 88, 133-140

https://doi.org/10.1016/j.compag.2012.07.008

[27] Spampinato, C., Palazzo, S. (2012) Hidden Markov models for detecting anomalous fish trajectories in underwater footage. En: IEEE International Workshop on Machine Learning for Signal Processing 2012, 23-26 Septiembre. Santander, España. https://doi.org/10.1109/mlsp.2012.6349768

[28] Matai, J., Kastner, R., Cutter, G. R., Demer, D. A. (2012). Automated techniques for detection and recognition of fishes using computer vision algorithms. En: Report of the National Marine Fisheries Service Automated Image Processing Workshop. 4-7 Septiembre. Seatle, W., USA.

[29] Boom, B. J., He, J., Palazzo, S., Huang, P. X., Beyan, C., Chou, H. M., Lin, F. P. et al. (2014) A research tool for long-term and continuous analysis of fish assemblage in coral-reefs using underwater camera footage. Ecological Informatics, 23, 83-97 https://doi.org/10.1016/j.ecoinf.2013.10.006

[30] Qin, H., Li, X., Liang, J., Peng, Y., and Zhang, C. (2016) DeepFish: Accurate underwater live fish recognition with a deep architecture. Neurocomputing, 187, 49-58 <u>https://doi.org/10.1016/j.neucom.2015.10.122</u>

[31] Huang, P. X., Boom, B. J., Fisher, R. B. (2014) GMM improves the reject option in hierarchical classification for fish recognition. En: IEEE Winter Conference on Applications of Computer Vision, 24-26 Marzo. Steamboat Spring, CO, USA. https://doi.org/10.1109/wacv.2014.6836076

[32] Boom, B. J., Huang, P. X., Beyan, C., Spampinato, C., Palazzo, S., He, J., Beauxis-Aussalet, E., Lin, S. I., Chou, H. M., Nadarajan, G., Chen-Burger, J., van-Ossenbruggen, J., Giordano, D., Hardman, L., Lin, F. P., Fisher, B. (2012) Long-term underwater camera surveillance for monitoring and analysis of fish populations, 11-15 Noviembre. Tsukuba, Japón.

[33] Jin, L., Liang, H. (2017) Deep learning for underwater image recognition in small sample size situations. En: IEEE OCEANS 2017, 19-22 Junio. Aberdeen, Escocia. <u>https://doi.org/10.1109/oceanse.2017.8084645</u>

[34] Ding, G., Song, Y., Guo, J., Feng, C., Li, G., He, B., Yan, T. (2017) Fish recognition using convolutional neural network. En: IEEE OCEANS 2017, 19-22 Junio. Aberdeen, Escocia.

[35] Abinaya, N. S., Susan, D., Kumar, R. (2021). Naive Bayesian fusion based deep learning networks for multisegmented classification of fishes in aquaculture industries. Ecological Informatics. 61, 101248

https://doi.org/10.1016/j.ecoinf.2021.101248

[36] Krizhevsky, A., Sutskever, I., Hinton, G. E. (2012) Imagenet classification with deep convolutional neural networks. Communications of the ACM. 60(6), 84-90 https://doi.org/10.1145/3065386

[37] Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A. A. (2017) Inception-v4, inception-resnet and the impact of residual connections on learning. En: Thirty-first AAAI conference on artificial intelligence, 4-9 Febrero. San Francisco, USA.

[38] He, K., Zhang, X., Ren, S., Sun, J. (2016) Deep residual learning for image recognition. En: Proceedings of the IEEE conference on computer vision and pattern recognition, 27-30 Junio. Las Vegas, USA.

https://doi.org/10.1109/cvpr.2016.90

[39] Girshick, R., Donahue, J., Darrell, T., Malik, J. (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, June 23-28. Columbus, USA. https://doi.org/10.1109/cvpr.2014.81

[40] Girshick, R. (2015) Fast r-cnn. En: Proceedings of the IEEE international conference on computer vision, 7-13 Diciembre. Santiago, Chile.

https://doi.org/10.1109/iccv.2015.169

[41] Ren S, He K, Girshick R, Sun J (2015) Faster r-cnn: Towards real-time object detection with region proposal networks. En: Advances in neural information processing systems, 7-12 Diciembre. Montreal, Canada.

https://doi.org/10.1109/tpami.2016.2577031

[42] Villon, S., Chaumont, M., Subsol, G., Villéger, S., Claverie, T., Mouillot, D. (2016) Coral reef fish detection and recognition in underwater videos by supervised machine learning: Comparison between Deep Learning and HOG + SVM methods. En: 17th International Conference on Advanced Concepts for Intelligent Vision Systems, 24-27 Octubre. Lecce, Italia.

[43] Villon, S., Mouillot, D., Chaumont, M., Darling, E. S., Subsol, G., Claverie, T., Villéger, S. (2018) A deep learning method for accurate and fast identification of coral reef fishes in underwater images. Ecological informatics, 48, 238-244.

[44] Tamou, A. B., Benzinou, A., Nasreddine, K., Ballihi, L. (2018) Underwater live fish recognition by deep learning. En: International Conference on Image and Signal Processing, 2-4 Julio. Cherbourg, Francia.

[45] Wechsler, H. (2014) Face recognition methods for uncontrolled settings. En: De Marsico, M. Nappi, M. Tistarelli, M. (ed). Face Recognition in Adverse Conditions. IGI Global, USA, pp 38-68.

https://doi.org/10.4018/978-1-4666-5966-7.ch003

[46] Jiang, G., Wang, S., Yu, M., Choi, T. Y., Kim, Y. D. (2004) New method of vision based vehicle detection and tracking in complicated background. En: IEEE Region 10 Conference TENCON, 21-24 Noviembre. Chiang Mai, Thailand.

https://doi.org/10.1109/tencon.2004.1414438

[47] MacLennan, D. N., Copland, P. J., Armstrong, E., Simmonds, E. J. (2004) Experiments on the discrimination of fish and seabed echoes, ICES Journal of Marine Science. 61(2), 201-210 <u>https://doi.org/10.1016/j.icesjms.2003.09.005</u>

[48] Lan, Y., Ji, Z., Gao, J., Yaowei, W. (2014) Robot fish detection based on a combination method of three-frame-difference and background subtraction. En: IEEE 26th Chinese Control and Decision Conference, 31 Mayo - 2 Junio. Changsha, China. http://dx.doi.org/10.1109/CCDC.2014.6852862

[49] Salman, A., Siddiqui, S. A., Shafait, F., Mian, A., Shortis, M. R., Khurshid, K., Ulges, A., Ulrich, S. (2019) Automatic fish detection in underwater videos by a deep neural network-based hybrid motion learning system. ICES Journal of Marine Science, 77(4), 1295-1307 http://dx.doi.org/10.1093/icesjms/fsz025

[50] Chuang, M. C., Hwang, J. N., Ye, J. H., Huang, S. C., Williams, K. (2016) Underwater fish tracking for moving cameras based on deformable multiple kernels. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 47(9), 2467-2477 <u>https://doi.org/10.1109/TSMC.2016.2523943</u> [51] Siddiqui, S. A., Salman, A., Malik, M. I., Shafait, F., Mian, A., Shortis, M. R., Harvey, E. S. (2018) Automatic fish species classification in underwater videos: exploiting pre-trained deep neural network models to compensate for limited labelled data, ICES Journal of Marine Science. 75(1), 374-389

https://doi.org/10.1093/icesjms/fsx109

[52] Allken, V., Handegard, N. O., Rosen, S., Schreyeck, T., Mahiout, T., Malde, K. (2019) Fish species identification using a convolutional neural network trained on synthetic data, ICES Journal of Marine Science. 76(1), 342-349 http://dx.doi.org/10.1093/icesjms/fsy147

[53] Redmon, J., Divvala, S., Girshick, R., Farhadi, A. (2016) You only look once: Unified, realtime object detection. En: IEEE Proceedings of the conference on computer vision and pattern recognition, 27-30 Junio. Las Vegas, USA.

https://doi.org/10.1109/cvpr.2016.91

[54] Chaves, L. C. T., Hall, J., Feitosa, J. L. L., Côté, I. M. (2016) Photo-identification as a simple tool for studying invasive lionfish Pterois volitans populations, Journal of fish biology. 88(2), 800-804

[55] Naddaf-Sh, M., Myler, H., Zargarzadeh, H. (2018) Design and implementation of an assistive real-time red lionfish detection system for AUV/ROVs. Complexity. <u>https://doi.org/10.1155/2018/5298294</u>

[56] Furlan, C., Boniface, A. (2018) Senior Project: Lionfish Detection System. Senior Project Report, Computer Engineering Department, California Polytechnic State University, California, USA. Disponible en <u>https://digitalcommons.calpoly.edu/cpesp/278/</u>

[57] Antaya, A. A., Peterson, E. H., Conroy, K. F., Ralph, T. V. (2019) Lionfish Bot 2.0. Final Report, Robotics Engineering Program, Worcester Polytechnic Institute, Massachusetts, USA. Disponible en <u>https://digitalcommons.wpi.edu/mqp-all/7046</u>

[58] Dickinson, J. L., Zuckerberg, B., Bonter, D. N. (2010) Citizen Science as an Ecological Research Tool: Challenges and Benefits. Annual Review of Ecology, Evolution, and Systematics 41, 149–172

https://doi.org/10.1146/annurev-ecolsys-102209-144636

[59] Scyphers, S. B., Powers, S. P., Akins, J. L., Drymon, J. M., Martin, C. W., Schobernd, Z. H., Schofield, P. J., Shipp, R. L., Switzer, T. S. (2015) The Role of Citizens in Detecting and Responding to a Rapid Marine Invasion, Conservation Letters. 8, 242–250 <u>https://doi.org/10.1111/conl.12127</u>

[60] Sale, P. F., Sharp, B. J. (1983) Correction for bias in visual transect censuses of coral reef fishes, Coral reefs. 2(1), 37-42

[61] Anderson, L. G., Chapman, J. K., Escontrela, D., Gough, C. L. (2017) The role of conservation volunteers in the detection, monitoring and management of invasive alien lionfish, Management of Biological Invasions. 8(4), 589-598

[62] Rogers, C. S., Gilnack, M., Fitz III, H. C. (1983) Monitoring of coral reefs with linear transects: a study of storm damage, Journal of Experimental Marine Biology and Ecology. 66(3), 285-300

[63] Lee, S., Buddo, D. S., Aiken, K. A. (2012) Habitat preference in the invasive lionfish (*Pterois volitans/miles*) in Discovery Bay, Jamaica: use of GIS in management strategies. En:
Proceedings of the 64th Gulf and Caribbean Fisheries Institute, 31 Octubre – 5 Noviembre.
Puerto Morelos, México.

[64] Frazer, T. K., Jacoby, C. A., Edwards, M. A., Barry, S. C., Manfrino, C. M. (2012) Coping with the lionfish invasion: can targeted removals yield beneficial effects?, Reviews in Fisheries Science. 20(4), 185-191

[65] Green, S. J., Tamburello, N., Miller, S. E., Akins, J. L., Côté, I. M. (2013) Habitat complexity and fish size affect the detection of Indo-Pacific lionfish on invaded coral reefs, Coral reefs. 32(2), 413-421

[66] Skewes, T., Dennis, D., Koutsoukos, A., Haywood, M., Wassenberg, T., Austin, M. (2002) Research for the sustainable use of beche-de-mer resources in the Torres Strait. Annual report, CSIRO Division of Marine Research, Cleveland, Australia. Disponible en https://www.academia.edu/download/42598777/r423.pdf

[67] Green, S. J. (2012) Monitoring: An Essential Action. En: Morris, Jr. J. A. (ed). Invasive Lionfish: A Guide to Control and Management. Gulf and Caribbean Fisheries Institute Special.Special Publication Series Number 1, Marathon, Florida, USA, pp 51–68

[68] Jacobi, M. (2015) Autonomous inspection of underwater structures, Robotics and Autonomous Systems. 67, 80–86 https://doi.org/10.1016/j.robot.2014.10.006

[69] Hover, F. S., Eustice, R. M., Kim, A., Englot, B., Johannsson, H., Kaess, M. Leonard, J. J. (2012). Advanced perception, navigation and planning for autonomous in-water ship hull inspection, The International Journal of Robotics Research. 31(12), 1445-1464 https://doi.org/10.1177/0278364912461059

[70] Evans, J., Redmond, P., Plakas, C., Hamilton, K. Lane, D. (2003) Autonomous docking for Intervention-AUVs using sonar and video-based real-time 3D pose estimation. En: IEEE Oceans 2003. Proceedings, 22-26 Septiembre, San Diego, CA, USA.

[71] De Novi, G., Melchiorri, C., Garcia, J. C., Sanz, P. J., Ridao, P. Oliver, G. (2009) A new approach for a reconfigurable autonomous underwater vehicle for intervention. En: IEEE 3rd Systems conference, 23-26 Marzo, Vancouver, BC, Canada.

[72] Prats, M., Ribas, D., Palomeras, N., García, J. C., Nannen, V., Wirth, S., Fernandez, J. J., Beltran, J. P., Campos, R., Ridao, P. Sanz, P. J. (2012) Reconfigurable AUV for intervention missions: a case study on underwater object recovery, Intelligent Service Robotics. 5(1), 19-31 https://doi.org/10.1007/s11370-011-0101-z

[73] Rojas, J. S. C. (2019) Vehículos Submarinos Autónomos, Pädi Boletín Científico de Ciencias Básicas e Ingenierías del ICBI. 7(Especial), 60-70 https://doi.org/10.29057/icbi.v7iEspecial.4432

[74] Van, M., Bui, D. H. P., Do, Q. T., Huynh, T. T., Lee, S. D., Choi, H. S. (2020) Study on dynamic behavior of unmanned surface vehicle-linked unmanned underwater vehicle system for underwater exploration, Sensors. 20(5), 1329

https://doi.org/10.3390/s20051329

[75] McLean, D. L., Parsons, M. J., Gates, A. R., Benfield, M. C., Bond, T., Booth, D. J., ..., & Jones, D. O. (2020) Enhancing the scientific value of industry remotely operated vehicles (ROVs) in our oceans, Frontiers in Marine Science. 7, 220 <u>https://doi.org/10.3389/fmars.2020.00220</u> [76] Huvenne, V. A., Robert, K., Marsh, L., Iacono, C. L., Le Bas, T., Wynn, R. B. (2018) Rovs and auvs. En: Micallef, A., Krastel, S., Savini, A. (ed). Submarine Geomorphology. Springer Geology, Cham, Suiza, pp 93-108

[77] Bonin-Font, F., Burguera, A., Oliver, G. (2011). Imaging systems for advanced underwater vehicles, Journal of Maritime Research. 8(1), 65-86

[78] Guillén, J. (2011) Visión Artificial Aplicada en Vehículos Autónomos Submarinos. Artículo universitario, Universidad Rey Juan Carlos de Madrid, España.

[79] Williams, S., Pizarro, O., Jakuba, M., Johnson, C., Barrett, N., Babcock, R., Kendrick, G., Steinberg, P., Heyward, A., Doherty, P., Mahon, I., Johnson-Roberson, M., Steinberg, D., Friedman, A. (2012) Monitoring of Benthic Reference Sites: Using an Autonomous Underwater Vehicle, IEEE Robotic Automation Magazine. 19, 73-84

[80] Paull, L., Saeedi, S., Seto, M., Li, H. (2014) AUV navigation and localization: A review, IEEE Journal of Oceanic Engineering. 39(1), 131-149

[81] Davis, R. E., Eriksen, C. C. Jones, C. P. (2002) Autonomous buoyancy-driven underwater gliders. En: G. Griffiths, (ed). The technology and applications of autonomous underwater vehicles, Taylor and Francis, London, pp 37-58

[82] Marani, G., Choi, S. K., Yuh, J. (2009) Underwater autonomous manipulation for intervention missions AUVs, Ocean Engineering. 36(1), 15-23

[83] Antonelli, G., Fossen, T. I. Yoerger, D. R., 2008. Underwater Robotics. In: Siciliano, B., Khatib, O. (ed). Springer Handbook of Robotics. Springer, Berlin, Heidelberg, pp. 987–1008

[84] Abreu, P. C., Botelho, J., Góis, P., Pascoal, A., Ribeiro, J., Ribeiro, M. Silva, H. (2016) The MEDUSA class of autonomous marine vehicles and their role in EU projects. En: IEEE OCEANS 2016, 10-13 Abril. Shanghai, China.

[85] Caccia, M., Bibuli, M., Bono, R., Bruzzone, G. (2008) Basic navigation, guidance and control of an unmanned surface vehicle, Autonomous Robots. 25(4), 349-365

[86] Del-Rio-Rivera, F., Ramírez-Rivera, V. M., Donaire, A., Ferguson, J. (2020) Robust Trajectory Tracking Control for Fully Actuated Marine Surface Vehicle, IEEE Access. 8, 223897-223904

https://doi.org/10.1109/ACCESS.2020.3042091

[87] Foster, S. D., Hosack, G. R., Hill, N. A., Barrett, N. S., Lucieer, V. L. (2014) Choosing between strategies for designing surveys: autonomous underwater vehicles, Methods Ecology and Evolution. 5, 287-297

[88] Ferrari, R., Bryson, M., Bridge, T., Hustache, J., Williams, S. B., Byrne, M., Figueira, W. (2016) Quantifying the response of structural complexity and community composition to environmental change in marine communities, Global Change Biology. 22, 1965-1975

[89] Morris, K. J., Bett, B. J., Durden, J. M., Benoist, N. M., Huvenne, V. A., Jones, D. O., Robert, K., Ichino, M. C., Wolff, G. A., Ruhl, H. A. (2016) Landscape-scale spatial heterogeneity in phytodetrital cover and megafauna biomass in the abyss links to modest topographic variation, Scientific Reports. 6(1), 1-10

https://doi.org/10.1038/srep34080

[90] Anderson, E. (2014) Bio-inspired robotic fish with vision based target tracking. Tesis de maestría. Purdue University, USA.

[91] RSE: Robots in Service of the Environment | Lionfish Robot (2019) Disponible en <u>https://robotsise.org</u> Consultado: 23 de Junio 2021

[92] Euler, L. (1741) Solutio problematis ad geometriam situs pertinentis. Commentarii academiae scientiarum Petropolitanae, 128-140.

[93] Guan, M. (1962) Graphic programming using odd and even points. Chinese Math., 1, 237-277.

[94] Orloff, C. S. (1974) A fundamental problem in vehicle routing. Networks, 4(1), 35-64.

[95] Rodríguez, A. M. V., Martín, F. A., Couso, F. G., González, M. L., Vera, G. P., Martín, C. V. (2014) Teoría de grafos. Ejercicios y problemas resueltos. Ediciones Paraninfo, SA. España.

[96] EEA W-Team (2021) Greenhouse gas data - Emissions share by sector in EU28. <u>http://www.eea.europa.eu/data-and-maps/data/data-viewers/greenhouse-gases-viewer</u> Consultado: 23 de noviembre de 2021 [97] Wen, M., Linde, E., Ropke, S., Mirchandani, P., Larsen, A. (2016) An adaptive large neighborhood search heuristic for the electric vehicle scheduling problem, Computers & Operations Research, 76, 73–83

[98] Villalobos, A. R. (2010) Grafos-Software para la construcción, edición y análisis de grafos.Bubok Publishing S. L. España.

[99] Stockdale, M. L. (2011). El problema del viajante: un algoritmo heurístico y una aplicación.Tesis de Licenciatura. Universidad de Buenos Aires. Argentina.

[100] Robinson, J. (1949) On the Hamiltonian game (a traveling salesman problem) (No. RAND/RM-303). RAND PROJECT AIR FORCE ARLINGTON VA.

[101] Roberti, R., Wen, M. (2016) The electric traveling salesman problem with time windows. Transportation Research Part E: Logistics and Transportation Review. 89, 32-52

[102] Doppstadt, C., Koberstein, A., Vigo, D. (2016) The hybrid electric vehicle-traveling salesman problem. European Journal of Operational Research. 253(3), 825-842

[103] Lu, C. C., Yan, S., Huang, Y. W. (2018) Optimal scheduling of a taxi fleet with mixed electric and gasoline vehicles to service advance reservations. Transportation Research Part C: Emerging Technologies. 93, 479-500

[104] Goeke, D., Schneider, M. (2015) Routing a mixed fleet of electric and conventional vehicles. European Journal of Operational Research. 245(1), 81-99

[105] Demir, E., Bektaş, T., Laporte, G. (2014) The bi-objective pollution-routing problem. European Journal of Operational Research. 232(3), 464-478

[106] Macrina, G., Laporte, G., Guerriero, F., Pugliese, L. D. P. (2019) An energy-efficient greenvehicle routing problem with mixed vehicle fleet, partial battery recharging and time windows. European Journal of Operational Research, 276(3), 971-982

[107] Demir, E., Bektaş, T., Laporte, G. (2012) An adaptive large neighborhood search heuristic for the pollution-routing problem. European Journal of Operational Research. 223(2), 346-359

[108] Barco, J., Guerra, A., Munoz, L., Quijano, N. (2013) Optimal routing and scheduling of charge for electric vehicles: A case study. Mathematical Problems in Engineering. <u>https://doi.org/10.1155/2017/8509783</u> [109] Amiri, S. S., Jadid, S., Saboori, H. (2018) Multi-objective optimum charging management of electric vehicles through battery swapping stations. Energy, 165, 549-562

[110] Mirmohammadi, S. H., Babaee Tirkolaee, E., Goli, A., Dehnavi-Arani, S. (2017) The periodic green vehicle routing problem with considering of time-dependent urban traffic and time windows, International Journal of Optimization in Civil Engineering. *7*(1), 143-156

[111] Mancini, S. (2017) The hybrid vehicle routing problem, Transportation Research Part C: Emerging Technologies. 78, 1-12

[112] Robotic Operating System <u>http://wiki.ros.org/noetic</u> Consultado: 23 de noviembre de 2021

[113] Gazebo Simulator http://gazebosim.org/ Consultado: 23 de noviembre de 2021

[114] https://opencv.org/opencv-3-0/ Consultado: 23 de noviembre de 2021

[115] https://www.ibm.com/docs/es/icos Consultado: 23 de noviembre de 2021

[116] Dantzig, G., Fulkerson, R., Johnson, S. (1954) Solution of a large-scale traveling-salesman problem. Journal of the operations research society of America, 2(4), 393-410

[117] <u>https://www.ibm.com/docs/es/icos/12.10.0?topic=mc-what-is-cplex</u> Consultado: 23 de noviembre de 2021

[118] Gonçalves, F., Cardoso, S. R., Relvas, S., Barbosa-Póvoa, A. P. F. D. (2011) Optimization of a distribution network using electric vehicles: A VRP problem. En: Proceedings of the IO2011-15 Congresso da associação Portuguesa de Investigação Operacional. 18-20 Abril. Coimbra, Portugal.

[119] Erdo^{*}gan, S., Miller-Hooks, E. (2012) A green vehicle routing problem," Transportation Research Part E: Logistics and Transportation Review. 48(1), 100–114

[120] A. Omidvar and R. Tavakkoli-Moghaddam, Sustainable vehicle routing: Strategies for congestion management and refueling scheduling, En: Proceedings of the IEEE International Energy Conference (ENERGYCON '12). 19-12 Septiembre. Florencia, Italia,

[121] Schneider, M., Stenger, A., Goeke, D. (2014) Te electric vehiclerouting problem with time windows and recharging stations," Transportation Science. 48(4), 500–520

[122] Keskin, M., Çatay, B. (2006) Partial recharge strategies for the electric vehicle routing problem with time windows, Transportation Research Part C: Emerging Technologies. 65, 111–127

[123] Padberg, M., Rinaldi, G. (1991) A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems, SIAM review. 33(1), 60-100

[124] Singh, A. (2019) Feature Engineering for Images: A Valuable Introduction to the HOG Feature Descriptor. Disponible en <u>https://www.analyticsvidhya.com/blog/2019/09/feature-engineering-images-introduction-hog-feature-descriptor/</u> Consultado: 08 de julio de 2021

[125] Dadi, H. S., & Pillutla, G. M. (2016) Improved face recognition rate using HOG features and SVM classifier. IOSR Journal of Electronics and Communication Engineering, 11(4), 34-44.

[126] Kumar, K. V. V., Kishore, P. V. V. (2017) Indian classical dance mudra classification using HOG features and SVM classifier. International Journal of Electrical & Computer Engineering. 7(5), 2088-8708

[127] Wang, Y., Zhu, X., & Wu, B. (2019) Automatic detection of individual oil palm trees from UAV images using HOG features and an SVM classifier. *International Journal of Remote Sensing*, 40(19), 7356-7370.

[128] Redmon, J., & Farhadi, A. (2017) YOLO9000: better, faster, stronger. In Proceedings of the IEEE conference on computer vision and pattern recognition. 7263-7271

[129] Redmon, J., & Farhadi, A. (2018). Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767.

[130] Bochkovskiy A, Wang CY, Liao HYM (2020) Yolov4: Optimal speed and accuracy of object detection. Computer Vision and Pattern Recognition, arXiv:2004.10934v1. Retrieved from https://arxiv.org/abs/2004.10934

[131] Página oficial de la base de datos COCO <u>https://cocodataset.org</u> Consultado: 23 de noviembre de 2021

[132] Página oficial de Nvidia- TensorRT <u>https://developer.nvidia.com/tensorrt</u> Consultado: 23 de noviembre de 2021

[133] Página oficial de Tensorflow <u>https://www.tensorflow.org</u> Consultado: 23 de noviembre de 2021

[134] Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, Andreeto M, Adam H (2017) MobileNets: Efficient convolutional neural networks for mobile vision applications. arXiv:1704.04861. Retrieved from https://arxiv.org/abs/1704.04861

[135] Chiu YC, Tsai CY, Ruan MD, Shen GY, Lee TT (2020) Mobilenet-SSDv2: An improved object detection model for embedded systems. In: IEEE 2020 International Conference on System Science and Engineering (ICSSE), Kagawa, Japan, July 5-8, 2020, pp. 1-5

[136] Página oficial de Pytorch https://pytorch.org/ Consultado: 23 de noviembre de 2021

[137]JetsonNano :DeepLearningInferenceBenchmarckshttps://developer.nvidia.com/embedded/jetson-nano-dl-inference-benchmarks.Consultado:23de noviembre de 2021

[138] Sistema de reconocimiento YOLOv4 <u>https://github.com/AlexeyAB/darknet#yolo-v4-in-other-frameworks</u> Consultado: 23 de noviembre de 2021

[139] Sistema de reconocimiento SSD-MobileNet-v2 <u>https://github.com/dusty-nv/jetson-inference</u> Consultado: 23 de noviembre de 2021

[140] Tzutalin (2015) Labellmg. Git code <u>https://github.com/tzutalin/labellmg</u> Consultado: 23 de noviembre de 2021

[141] Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles, Ground Vehicle Standard J3016-201806, Published 2018-06-15 by SAE International in United States.

[142] Ignacio, L. C., Victor, R. R., Del Rio, R., Pascoal, A. (2019) Optimized design of an autonomous underwater vehicle, for exploration in the Caribbean Sea, Ocean Engineering. <u>https://doi.org/10.1016/j.oceaneng.2019.106184</u>

[143] Vásquez, L. O. P., Ramírez, V. M., Thanapalan, K. (2020) A comparison of energy management system for a DC microgrid, Applied Sciences. <u>https://doi.org/10.3390/app10031071</u>

[144] González, I., Sánchez, A., Langarica-Cordoba, D., Yanine-Misleh, F., & Ramirez, V. (2021)
 A PI+ Sliding-Mode Controller Based on the Discontinuous Conduction Mode for an Unidirectional Buck–Boost Converter with Electric Vehicle Applications, Energies.
 <u>https://doi.org/10.3390/en14206785</u>

[145] Martínez-González, Á. T., Ramírez-Rivera, V. M., Caballero-Vázquez, J. A., Jáuregui, D.
A. G. (2021) Deep learning algorithm as a strategy for detection an invasive species in uncontrolled environment, Reviews in Fish Biology and Fisheries.
https://doi.org/10.1007/s11160-021-09667-7

ANEXOS

ANEXO 1

Código para obtener la profundidad de las imágenes izquierda y derecha obtenidas de las cámaras virtuales.

import numpy as np import cv2 from matplotlib import pyplot as plt from PIL import Image

left_image = cv2.imread('Imagenes/imagenR32.png', cv2.IMREAD_GRAYSCALE)
right_image = cv2.imread('Imagenes/imagenL32.png', cv2.IMREAD_GRAYSCALE)

ANEXO 2

Código para la generación de los puntos de interés

import numpy as np import cv2 import math import csv

print (cont) |x1=|y1=|x2=|y2=|x3=|y3=|x4=|y4=|x5=|y5=3000Lim=250 Dila1=38 Dila2=48 RadioD=260 Tipo_Monit=1 # Si es 1, Monitoreo minucioso; si es otro, monitoreo libre Num Lineas=35 # Sólo 3, 4 o 5 o 35 mezcla # Definir valores de las lineas if Num Lineas==3: Lin1= 604 Lin2= 1208 Lin3= 1812 Lin4= 1208 Lin5 = 604Data = [(Lin3,Lin3),(Lin2,Lin3),(Lin1,Lin3),(Lin3,Lin2),(Lin2,Lin2),(Lin1,Lin2), (Lin3,Lin1),(Lin2,Lin1),(Lin1,Lin1)] elif Num_Lineas==4: Lin1= 490 Lin2= 980 Lin3= 1470 Lin4= 1960 Lin5= 490 Data = [(Lin4,Lin4),(Lin3,Lin4),(Lin2,Lin4),(Lin1,Lin4),(Lin4,Lin3),(Lin3,Lin3),(Lin2,Lin3),(Lin1,Lin3), (Lin4,Lin2),(Lin3,Lin2),(Lin2,Lin2),(Lin1,Lin2), (Lin4,Lin1),(Lin3,Lin1),(Lin2,Lin1),(Lin1,Lin1)] elif Num_Lineas==5: Lin1 = 410Lin2 = 820Lin3= 1230 Lin4 = 1640Lin5 = 2050Data = [(Lin5,Lin5),(Lin4,Lin5),(Lin3,Lin5),(Lin2,Lin5),(Lin1,Lin5),(Lin5,Lin4),(Lin4,Lin4),(Lin3,Lin4),(Lin2,Lin4),(Lin1,Lin4), (Lin5,Lin3),(Lin4,Lin3),(Lin3,Lin3),(Lin2,Lin3),(Lin1,Lin3), (Lin5,Lin2),(Lin4,Lin2),(Lin3,Lin2),(Lin2,Lin2),(Lin1,Lin2), (Lin5,Lin1),(Lin4,Lin1),(Lin3,Lin1),(Lin2,Lin1),(Lin1,Lin1)] elif Num Lineas==35: Lin1 = 410

```
Lin2= 820
      Lin3= 1230
      Lin4= 1640
      Lin5= 2050
      Data = [(1912, 1912), (1208, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 1912), (504, 
                 (1912,1208),(1208,1208),(504,1208),
                 (1912,504),(1208,504),(504,504)]
                                                    # Quitar los lados blancos que no se usarán
img = cv2.rectangle(Image, (47,28), (153,2436), (0,0,0), 110)
img = cv2.rectangle(Image, (2429,8), (2447,2455), (120,0,0), 35)
img = cv2.rectangle(Image, (206, 14), (2410, 36), (90, 0, 0), 30)
img = cv2.rectangle(Image, (206,2413), (2410,2448), (50,0,0), 34)
                                                    # Dilatar la imagen
kernel = np.ones((5,5),np.uint8)
Dilata1 = cv2.dilate(img,kernel,iterations = Dila1)
Dilata2 = cv2.dilate(img,kernel,iterations = Dila2)
                                                    #Binarización
ret,thresh1 = cv2.threshold(Dilata1,127,255,cv2.THRESH_BINARY)
ret,thresh2 = cv2.threshold(Dilata2,127,255,cv2.THRESH_BINARY)
ImNueva=np.zeros((2464,2464))
ImNegra = ImNueva.astype(np.uint8)
ImLinea = ImNueva.astype(np.uint8)
                                                    #Obtención de contornos
contoursU,
                                            hierarchy
                                                                                     =
                                                                                                          cv2.findContours(thresh1,
                                                                                                                                                                                           cv2.RETR_TREE,
cv2.CHAIN_APPROX_NONE)
contours,
                                         hierarchy
                                                                                                          cv2.findContours(thresh2,
                                                                                                                                                                                           cv2.RETR TREE,
                                                                                   =
cv2.CHAIN_APPROX_NONE)
                                                    #Procesamiento para encontrar puntos
cont = 0
listC=[]
for c in contours:
      area = cv2.contourArea(c)
      M = cv2.moments(c)
      cx=-1
      cy=-1
      if (M['m00']!=0):
            cx = int(M['m10']/M['m00'])
            cy= int(M['m01']/M['m00'])
            co_centro = (cx, cy)
      if area <= 50000:
            listC.append(c)
            PuntosC.append(co_centro)
            PuntosCF.append(co centro)
      else:
```
```
listF.append(c)
  cont = cont+1
cont = 0
for c in listF:
  tamaño=len(c)
  for t in range(tamaño):
     c2=c[t,0]
     c3=c2[0]
     c4=c2[1]
     if c3 == Lin1 or c4 == Lin1 :
       if c3 == Lin1 and math.sqrt((ly1 - c4)**2) <=Lim:
          contmño= contmño + 1
       elif c4 == Lin1 and math.sqrt((Ix1 - c3)^{**2}) <=Lim:
          contmño= contmño + 1
       else:
          p = (c3, c4)
          PuntosF.append(p)
          PuntosCF.append(p)
          cont2.append(cont)
          |x1=c3|
          lv1 = c4
     elif c3 == Lin2 or c4 == Lin2:
       if c3 == Lin2 and math.sqrt((ly2 - c4)**2) <=Lim:
          contmño= contmño + 1
       elif c4 == Lin2 and math.sqrt((Ix2 - c3)^{**}2) <=Lim:
          contmño= contmño + 1
       else:
          p = (c3, c4)
          PuntosF.append(p)
          PuntosCF.append(p)
          cont2.append(cont)
          Ix2 = c3
          ly2 = c4
     elif c3 == Lin3 or c4 == Lin3:
       if c3 == Lin3 and math.sqrt((ly3 - c4)**2) <=Lim:
          contmño= contmño + 1
       elif c4 == Lin3 and math.sqrt((Ix3 - c3)^{**2}) <=Lim:
          contmño= contmño + 1
       else:
          p = (c3, c4)
          PuntosF.append(p)
          PuntosCF.append(p)
          cont2.append(cont)
```

```
x_3 = c_3
          ly3 = c4
     elif c3 == Lin4 or c4 == Lin4:
       if c3 == Lin4 and math.sqrt((ly4 - c4)**2) <=Lim:
          contmño= contmño + 1
       elif c4 == Lin4 and math.sqrt((Ix4 - c3)^{**2}) <=Lim:
          contmño= contmño + 1
       else:
          p = (c3, c4)
          PuntosF.append(p)
          PuntosCF.append(p)
          cont2.append(cont)
          Ix4 = c3
          V4 = c4
     elif c3 == Lin5 or c4 == Lin5:
       if c3 == Lin5 and math.sqrt((ly5 - c4)**2) <=Lim:
          contmño= contmño + 1
       elif c4 == Lin5 and math.sqrt((Ix5 - c3)^{**2}) <=Lim:
          contmño= contmño + 1
       else:
          p = (c3, c4)
          PuntosF.append(p)
          PuntosCF.append(p)
          cont2.append(cont)
          Ix5 = c3
          ly5 = c4
  cont=cont+1
       #Procesando puntos predefinidos. Si el punto está en pixel negro, se toma.
cont=0
contS=0
if Tipo Monit == 1:
  for n in Data:
     i=n[1]
     j=n[0]
     if thresh1[i,j] == 0:
       Punt_pre.append(n)
       PuntosCF.append(n)
for c in contoursU:
  area = cv2.contourArea(c)
  M = cv2.moments(c)
  cx=-1
  cy=-1
  if (M['m00']!=0):
     cx= int(M['m10']/M['m00'])
```

```
cy= int(M['m01']/M['m00'])
     co_centro = (cx, cy)
  if area <= 50000:
     PuntosCF.append(co_centro)
  cont = cont+1
#print (PuntosC)
cont = 0
e = len(PuntosCF)
while e != 0:
  for x in PuntosCF:
     TupIn.append(x)
     for t in TupIn:
       if x == t or contS in TupS:
          if cont not in Tup1:
            Tup1.append(cont)
       else:
          a = x[0]-t[0]
          b = x[1]-t[1]
          r = math.sqrt(b^{**}2 + a^{**}2)
          if r <= RadioD:
            a=t[0]+(a/2)
            b=t[1]+(b/2)
            pp=(a,b)
            TupN.append(pp)
            TupS.append(cont)
            TupS.append(contS)
          else:
            if cont not in Tup1:
               Tup1.append(cont)
       contS=contS+1
     contS=0
     cont=cont+1
  #print(len(PuntosCF))
  #print(len(TupIn))
  s= len(TupS)
  if s == 0:
     break
  PuntosCF=[]
  for g in Tup1:
     if g not in TupS:
```

```
k=TupIn[g]
       PuntosCF.append(k)
  for c in TupN:
     PuntosCF.append(c)
  TupIn,TupN,Tup1,TupS=[],[],[],[]
  cont=0
              # Guardado de las coordenadas de los puntos de interés en archivo .csv
#myFile = open('ResImage/Archivo ML3L.csv', 'w')
myFile = open('ResImage9/Archivo35L.csv', 'w')
                                                                       #
with myFile:
  writer = csv.writer(myFile)
  writer.writerow(PuntosCF)
  print("Writing complete")
              # Para visualizar imágenes resultantes
cv2.drawContours(ImNegra, listF, -1, color, thickness)
cv2.drawContours(ImNegra, listC, -1, color, thickness)
for np in PuntosCF:
  npj=int(np[1])
  npi=int(np[0])
  cv2.circle(ImLinea, (npi,npj), 6, (255,0,0), 6)
  cv2.circle(Image2, ((npi-100),(npj)), 8, (255,0,0), 8)
                     # Guardado de imágenes
cv2.imwrite('ResImage9/FormasPuntosFin35L.png',ImLinea)
cv2.imwrite('ResImage9/ImgFin35L.png',Image2)
#cv2.imwrite('ResImage32/ML_FormasPuntosFin3L.png',ImNegra)
#cv2.imwrite('ResImage32/Monit Libre3L.png',Image2)
```

Programación para la solución de rutas óptimas por medio de método TSP de la librería Cplex de IBM.

TSP Final #Comenzaremos importanto las librerias necesarias. import numpy as np import matplotlib.pyplot as plt from docplex.mp.model import Model import docplex.mp.solution as Solucion import csv import math

```
conta=0
inicio= 7
fin=7
Data, Distan, Distan2 = [],[],[]
```

```
coordex,coordey,coordex2,coordey2,arcosT = [],[],[],[],[]
Titulo="Camino óptimo y distancia - TSP - Im24_Mm_5L"
Titulo2="Im24 Mm 5L.png"
with open('CSVs24/Archivo5L.csv') as File:
                                                  skipinitialspace=True)#,
  reader
                           csv.reader(File,
                                                                                  delimiter=',',
                 =
quoting=csv.QUOTE_MINIMAL)#quotechar=',',
  for row in reader:
     Data.append(row)
coords = Data[0]
coordey.append(0)
coordex.append(inicio)
if len(coords) <=21:
  ciclos=1
else: ciclos=2
gP = 80/2464
Prof = 8 \# (metros)
       # Corrección de coordenadas de pixeles a distancia en metros
for e in coords:
  e=eval(e)
  Apx = e[0] - 1232
  Apy = e[1] - 1232
  if ciclos==1:
     Dx=math.tan((gP*Apx)*math.pi/180) * Prof
     Dx=7 + Dx
     coordex.append(Dx)
     Dy=math.tan((gP*Apy)*math.pi/180) * Prof
     Dy= 7 - Dy
     coordey.append(Dy)
  elif ciclos==2:
     if Apy \geq 0:
       Dx=math.tan((gP*Apx)*math.pi/180) * Prof
       Dx=7 + Dx
       coordex.append(Dx)
       Dy=math.tan((gP*Apy)*math.pi/180) * Prof
       Dy= 7 - Dy
       coordey.append(Dy)
     else:
       Dx=math.tan((gP*Apx)*math.pi/180) * Prof
       Dx=7 + Dx
       coordex2.append(Dx)
       Dy=math.tan((gP*Apy)*math.pi/180) * Prof
       Dy= 7 - Dy
       coordey2.append(Dy)
```

```
# Los ciclos for están dentro de [], para crear listas con esa cantidad de datos.
Guardax=coordex + coordex2
Guarday=coordey + coordey2
while ciclos != 0:
  coordey.append(14)
  coordex.append(7)
  n=len(coordex)
  puntos=[i for i in range(n)] # Creamos puntos de la 0 a la n
  arcos =[(i,j) for i in puntos for j in puntos if i!=j if j!=0 if i!=n-1]
  x=coordex
  y=coordey
  plt.figure(figsize=(14,14))
  plt.scatter(x,y,color='blue')
  s=[]
  for m in range(len(coordex)):
     s_temp=[]
     s_temp.append("%.1f" %coordex[m])
     s_temp.append("%.1f" %coordey[m])
     s.append(s_temp)
  plt.xlabel("Distancia X")
  plt.ylabel("Distancia Y")
  plt.title("Ubicación de los puntos - Im3_Mm_5L")
  for m in range(len(coordex)):
     plt.annotate(str(s[m]), xy=(coordex[m],coordey[m]), xytext=(coordex[m]-0.5,coordey[m]-
0.5),
          color='purple')
  for m in range(len(coordex)):
     plt.annotate(str(m), xy=(coordex[m],coordey[m]), xytext=(coordex[m]+0.3,coordey[m]+0.3),
          color='red')
  plt.xlim((0,14))
  plt.ylim((-1,15))
  # plt.show()
#Ahora debemos crear un link entre los puntos, sus coordenadas y la distancia entre ellas.
#Los parentesis de llave significan que es un diccionario { }. De esta manera creamos un link
# entre los arcos y la distancia.
  distancia={(i, j): np.hypot(coordex[i] - coordex[j], coordey[i] - coordey[j]) for i,j in arcos}
# Creando el modelo en Cplex
  mdl=Model('TSP')
#Declaramos las variables a utilizar
#los nombre y forma de llamar las variables es propia de CPLEX.
```

```
x=mdl.binary_var_dict(arcos,name='x')
  d=mdl.continuous_var_dict(puntos,name='d')
# Funcion objetivo
  mdl.minimize(mdl.sum(distancia[i]*x[i] for i in arcos))
# Primero restricción sumatoria para todo i
  for c in puntos:
     if c!=n-1:
       mdl.add_constraint(mdl.sum(x[(i,j)] for i,j in arcos if i==c)==1,
               ctname='out_%d'%c)
# Segundo restricción sumatoria para todo j
  for c in puntos:
     if c!=0:
       mdl.add_constraint(mdl.sum(x[(i,j)] for i,j in arcos if j==c)==1,
               ctname='in_%d'%c)
#print(mdl.export_to_string())
# Por último restricción lógica
  for i,j in arcos:
     if j!=0:
       mdl.add_indicator(x[(i,j)],d[i]+1==d[j],
                name='order_(%d,_%d)'%(i, j))
     if i!=n-1:
       mdl.add_indicator(x[(i,j)],d[i]+1==d[j],
                name='order_(%d,_%d)'%(i, j))
  mdl.parameters.timelimit=240
  mdl.parameters.mip.strategy.branch=1
  mdl.parameters.mip.tolerances.mipgap=0.15
  solucion = mdl.solve(log_output=True)
  mdl.get solve status()
  solucion.display()
#Imprimiento la solución óptima.
  plt.figure(figsize=(14,14))
  plt.xlabel("Distancia X")
  plt.ylabel("Distancia Y")
  plt.title("Camino óptimo - TSP - Im3_Mm_5L")
  arcos_activos = [i for i in arcos if x[i].solution_value > 0.9]
  for i,j in arcos activos:
     plt.plot([coordex[i],coordex[i]],[coordey[i],coordey[i]],
         color='b', alpha=0.4, zorder=0)
  plt.scatter(x=coordex, y=coordey, color='blue', zorder=1)
  for m in range(len(coordex)):
     plt.annotate(str(s[m]), xy=(coordex[m],coordey[m]),
           xytext=(coordex[m]-0.5,coordey[m]-0.5), color='purple')
  for m in range(len(coordex)):
```

```
plt.annotate(str(m), xy=(coordex[m],coordey[m]),
           xytext=(coordex[m]+0.3,coordey[m]+0.3),color='red')
  plt.xlim((0,14))
  plt.ylim((-1,15))
 # plt.show()
  tam = len(arcos activos)
  for rev in arcos_activos:
     if tam in rev:
       r=rev[0]
       rx=coordex[r]
       ry=coordey[r]
       coordex2.insert(0,rx)
       coordey2.insert(0,ry)
        c=conta
     conta=conta + 1
  if ciclos == 2:
     tam1=tam
     r2=r
     arcos_activos.pop(c)
  if len(arcosT) != 0:
     for ch in arcos_activos:
        cx=ch[0] + (tam 1-1)
        cy=ch[1] + (tam 1-1)
       cxy=(cx,cy)
       arcosT.append(cxy)
     s1=arcosT[tam1-1]
     arcosT[tam1-1]=(r2,s1[1])
  else: arcosT= arcosT + (arcos_activos)
  ciclos = ciclos - 1
  coordex=coordex2
  coordey=coordey2
Guardax.append(7)
Guarday.append(14)
for i,j in arcosT:
  Dist=np.hypot(Guardax[i] - Guardax[j], Guarday[i] - Guarday[j])
  Distan.append(Dist)
plt.figure(figsize=(22,16))
plt.xlabel("Distancia X (m)",fontsize=16)
plt.ylabel("Distancia Y (m)",fontsize=16)
plt.title(Titulo, fontsize=20)
for i,j in arcosT:
  plt.plot([Guardax[i],Guardax[j]],[Guarday[i],Guarday[j]],
         color='b', alpha=0.6, zorder=0)
plt.scatter(x=Guardax, y=Guarday, color='blue', zorder=1)
```

```
for m in range(len(Distan)):
  plt.annotate("%.2f"%Distan[m], xy=(Guardax[m],Guarday[m]),
          xytext=(Guardax[m]-0.8,Guarday[m]-0.8), color='purple',fontsize=16)
for m in range(len(Guardax)):
  plt.annotate(str(m), xy=(Guardax[m],Guarday[m]),
          xytext=(Guardax[m]+0.2,Guarday[m]+0.2),color='red', fontsize=16)
lenD=len(Distan)
Distan2=Distan
Distan2.pop(0)
Distan2.pop(lenD-2)
DistanT= sum(Distan2)
print(DistanT)
plt.annotate("Distancia Total: %.2f m"%DistanT, xy=(Guardax[m],Guarday[m]),
          xytext=(9,-0.8), color='green', fontsize=18)
plt.xlim((0,14))
plt.ylim((-1,15))
plt.savefig(Titulo2, dpi=300)
#plt.show()
```

Resultados de las rutas óptimas de los 42 escenarios simulados para el monitoreo libre (Mol-AUV)



Op4_ML_5L

Ilustración 1. Impresión de ruta sobre escenario (1 a 6).



Op7_ML_5L



Op9_ML_5L

7.69



Op10_ML_5L



Op11_ML_5L



Op12_ML_5L



Op13_ML_5L







Op15_ML_5L



Ilustración 3. Impresión de ruta sobre escenario (13 a 18).

105





Distancia Total: 29.00 m Dip23_ML_5L



Op24_ML_5L

Op22_ML_5L

Ilustración 4. Impresión de ruta sobre escenario (19 a 24).



Op28_ML_5L Op29_ML_5L

Op30_ML_5L

Ilustración 5. Impresión de ruta sobre escenario (25 a 30).





Ilustración 7. Impresión de ruta sobre escenario (37 a 42).

Resultados de validación de los 5 modelos entrenados para el sistema de reconocimiento del pez león con YOLOv4.



Ilustración 8. Captura de gráfica de resultados de validación para el modelo 1, y resultado de validación en terminal del mejor desempeño del modelo.



Ilustración 9. Captura de gráfica de resultados de validación para el modelo 2, y resultado de validación en terminal del mejor desempeño del modelo.



Ilustración 10. Captura de gráfica de resultados de validación para el modelo 3, y resultado de validación en terminal del mejor desempeño del modelo.



Ilustración 11. Captura de gráfica de resultados de validación para el modelo 4, y resultado de validación en terminal del mejor desempeño del modelo.



Ilustración 12. Captura de gráfica de resultados de validación para el modelo 5, y resultado de validación en terminal del mejor desempeño del modelo.

Resultados del sistema de reconocimiento del pez león con YOLOv4



