



FACULTAD DE INGENIERÍA

DETECCIÓN Y SEGMENTACIÓN DE PECES UTILIZANDO
TÉCNICAS DE *DEEP LEARNING*

Tesis presentada por:
Walker Fernando Manrique Chalco

Para optar el Grado de:
Ingeniero de Software

Asesor:
Prof. Vicente Enrique Machaca Arceda

Arequipa - Perú

2023

21.9%Resultados del Análisis de los plagios del 2023-03-16 23:23 UTC
3112-3_3.pdf

Fecha: 2023-03-16 23:18 UTC

* Todas las fuentes 100 | Fuentes de internet 97 | Documentos propios 3

- ✓ [0] [core.ac.uk/download/pdf/80047395.pdf](#)
5.1% 85 resultados
- ✓ [1] [oa.upm.es/47784/1/TFM_JOSELUIS_GALARRAGA_CANIZARES.pdf](#)
5.4% 87 resultados
- ✓ [2] [ri.unlu.edu.ar/xmlui/bitstream/handle/rediunlu/842/Trabajo_Final_Licenciatura_Uria_Santiago_113311.pdf?sequence=1&isAllowed=y](#)
5.1% 81 resultados
- ✓ [3] [riunet.upv.es/bitstream/handle/10251/185962/Primo - PREDICCION DEL ENGAGEMENT EN INSTAGRAM APLICADO AL MUNDO DI](#)
4.3% 55 resultados
- ✓ [4] [eprints.ucm.es/55256/1/T41074.pdf](#)
3.4% 67 resultados
- ✓ [5] [www.ctrl.cinvestav.mx/~yuw/pdf/DoTesMLopez.pdf](#)
3.2% 65 resultados
- ✓ [6] [openaccess.uoc.edu/bitstream/10609/139006/6/jbustospelegriTFM0122memoria.pdf](#)
3.3% 59 resultados
- ✓ [7] [oa.upm.es/51441/1/TFM_OSCAR_PICAZO_MONTOYA.pdf](#)
3.0% 62 resultados
- ✓ [8] [www.juanbarrios.com/google-machine-learning/](#)
3.2% 48 resultados
- ✓ [9] [www.ptolomeo.unam.mx:8080/xmlui/bitstream/handle/132.248.52.100/15458/tesis.pdf?sequence=1&isAllowed=y](#)
2.6% 52 resultados
- ✓ [10] [oa.upm.es/63719/1/TFM_ALFONSO_HERNANDEZ_SILVA.pdf](#)
2.8% 49 resultados
- ✓ [11] [oa.upm.es/69291/1/TFG_JOSE_MIGUEL_SANCHEZ_SANCHEZ.pdf](#)
2.8% 46 resultados
- ✓ [12] [eprints.ucm.es/74259/1/natalia-casado-redes-neuronales-tfg.pdf](#)
2.7% 54 resultados
- ✓ [13] [riunet.upv.es/bitstream/handle/10251/109265/Torregrosa Lloret - Diseño de una segmentación automática de gliomas en imágenes de](#)
2.5% 45 resultados
- ✓ [14] [scielo.sld.cu/scielo.php?script=sci_arttext&pid=S2227-18992019000400118](#)
2.6% 40 resultados
- ✓ [15] [oa.upm.es/71437/1/TFG_IGNACIO_GOMEZ_VALVERDE.pdf](#)
2.2% 46 resultados
- ✓ [16] [oa.upm.es/53790/1/TFG_ADRIAN_FERNANDEZ_GALLARDO.pdf](#)
2.4% 43 resultados
- ✓ [17] [eprints.ucm.es/23444/1/ProyectoFinMasterPedroPablo.pdf](#)
2.3% 50 resultados
- ✓ [18] [es.wikipedia.org/wiki/Segmentación_\(procesamiento_de_imágenes\)](#)
2.2% 37 resultados
- ✓ [19] [www.redalyc.org/journal/5122/512261374010/html/](#)
1.9% 40 resultados
- ✓ [20] [biblus.us.es/bibing/proyectos/abreproy/93731/fichero/TFG-3731 MUÑIZ GARCÍA, LUIS.pdf](#)
2.1% 45 resultados
- ✓ [21] [revistas.ulima.edu.pe/index.php/CIIS/article/download/5507/5207/](#)
2.0% 40 resultados
- ✓ [22] [oa.upm.es/55751/1/TFM_VICTOR_DE_LA_FUENTE_CASTILLO.pdf](#)
1.7% 35 resultados
- ✓ [23] [openaccess.uoc.edu/bitstream/10609/95886/6/irodriguezgarTM0619memoria.pdf](#)
1.6% 37 resultados
- ✓ [24] [riunet.upv.es/bitstream/handle/10251/70104/53762075N_TFG_14732886778528883424789181367181.pdf?sequence=2](#)
1.6% 31 resultados

-
- [25] core.ac.uk/download/pdf/296326928.pdf
1.5% 30 resultados
1 documento con coincidencias exactas
-
- [27] portal.amelica.org/ameli/jatsRepo/606/6062590003/html/index.html
2.0% 31 resultados
-
- [28] portal.amelica.org/ameli/journal/602/6022798005/html/
1.6% 30 resultados
-
- [29] www.scielo.cl/scielo.php?pid=S0718-50062020000500233&script=sci_arttext
1.5% 32 resultados
-
- [30] sedici.unlp.edu.ar/bitstream/handle/10915/120856/Documento_completo.pdf-PDFA.pdf?sequence=1&isAllowed=y
1.4% 30 resultados
-
- [31] eprints.ucm.es/61386/1/1138495867-305939_Almodena_Outeda_Rodríguez_TFG_Almodena_Outeda_3940149_293736724.pdf
1.4% 29 resultados
-
- [32] www.intel.es/content/www/es/es/internet-of-things/computer-vision/convolutional-neural-networks.html
1.5% 25 resultados
-
- [33] es.wikipedia.org/wiki/Seguimiento_de_objetos
1.4% 24 resultados
-
- [34] sedici.unlp.edu.ar/bitstream/handle/10915/101507/Documento_completo.pdf-PDFA.pdf?sequence=1&isAllowed=y
1.5% 22 resultados
-
- [35] docplayer.es/125427609-Ana-maria-contreras-huertas-manuel-sebastian-leon-velandia.html
0.9% 27 resultados
-
- [36] core.ac.uk/download/pdf/226760502.pdf
1.1% 23 resultados
1 documento con coincidencias exactas
-
- [38] eprints.ucm.es/74255/1/RAMOS_MESA_84973_SERGIO_RAMOS_MESA_Autonomia_de_vehiculos_Follow_me_en_zonas_aeroportuar
1.1% 22 resultados
-
- [39] hmong.es/wiki/Object_detection
1.3% 15 resultados
-
- [40] repositorio.ucsp.edu.pe/bitstream/UCSP/15996/2/MUÑOZ_SALAS_GI_M1.pdf
1.0% 23 resultados
-
- [41] inteligenciaartificialhoy.com/que-son-las-redes-neuronales-en-el-aprendizaje-automatico-en-inteligencia-artificial/
1.3% 19 resultados
-
- [42] interactivechaos.com/es/wiki/machine-learning-vs-deep-learning
1.3% 21 resultados
-
- [43] www.ibm.com/ar-es/topics/supervised-learning
1.1% 16 resultados
-
- [44] riunet.upv.es/bitstream/handle/10251/174254/Aibar_Closed_Captions_Generador_de_subtitulos_automaticos_offline_empleando_un_moto
0.8% 14 resultados
-
- [45] 1library.co/document/1y972pdq-caracterizacion-automatica-especies-madera-mediante-tecnicas-clasificacion-imagenes.html
0.8% 16 resultados
-
- [46] aws.amazon.com/es/what-is/neural-network/
0.8% 14 resultados
-
- [47] es.linkedin.com/pulse/machine-learning-5-algoritmos-que-sirven-para-entrenar-
0.8% 15 resultados
-
- [48] 1library.co/article/arte-comprensión-semántica-escena.7qvn4x1z
0.6% 12 resultados
-
- [49] www.redalyc.org/journal/5122/512267931005/html/
0.6% 16 resultados
-
- [50] datascientest.com/es/machine-learning-definicion-funcionamiento-usos
0.6% 15 resultados
-
- [51] www.juanbarrios.com/la-matriz-de-confusion-y-sus-metricas/
0.7% 16 resultados
-
- [52] learn.microsoft.com/es-es/dotnet/machine-learning/resources/tasks
0.5% 11 resultados

- ✓ [53]  repositorio.esan.edu.pe/bitstream/handle/20.500.12640/1743/2019_MAM_17-2_05_T.pdf?sequence=6&isAllowed=y
0.5% 9 resultados

- ✓ [54]  es.wikipedia.org/wiki/Aprendizaje_de_clasificación
0.8% 13 resultados

- ✓ [55]  www.ibm.com/mx-es/cloud/learn/neural-networks
0.6% 8 resultados

- ✓ [56]  "Borrador - Clima Organizacional Levantando observaciones PRECOFERT -GAA- (1) (2).docx" fechado del 2022-09-22
0.5% 12 resultados

- ✓ [57]  interactivechaos.com/es/wiki/machine-learning
0.6% 12 resultados

- ✓ [58]  www.juanbarrios.com/evaluando-los-algoritmos-de-clasificacion/
0.6% 8 resultados

- ✓ [59]  www.ibm.com/es-es/topics/convolutional-neural-networks
0.6% 9 resultados

- ✓ [60]  repositorio.ucv.edu.pe/bitstream/handle/20.500.12692/65973/Valverde_RRA-SD.pdf?sequence=1
0.4% 12 resultados

- ✓ [61]  es.mathworks.com/discovery/convolutional-neural-network-matlab.html
0.4% 11 resultados

- ✓ [62]  aprendeia.com/todo-sobre-aprendizaje-no-supervisado-en-machine-learning/
0.4% 11 resultados

- ✓ [63]  aprendeia.com/evaluando-el-error-en-los-modelos-de-clasificacion-machine-learning/
0.5% 6 resultados

- ✓ [64]  repositorio.ug.edu.ec/bitstream/redug/60147/1/BINGQ-IQ-22P06.pdf
0.4% 9 resultados

- ✓ [65]  repositorio.upse.edu.ec/bitstream/46000/5465/1/UPSE-TAE-2020-0065.pdf
0.4% 10 resultados

- ✓ [66]  es.wikipedia.org/wiki/Detección_de_objetos
0.4% 5 resultados

- ✓ [67]  addi.ehu.es/bitstream/handle/10810/25348/TESIS_CAMINOS_CASADO_LEIRE.pdf?isAllowed=y&sequence=1
0.3% 12 resultados

- ✓ [68]  es.linkedin.com/pulse/aprendizaje-profundo-en-sistemas-de-recomendación-parra-santander
0.4% 11 resultados

- ✓ [69]  revistasdex.uchile.cl/index.php/bits/article/download/2777/2710/10190
0.4% 11 resultados

- ✓ [70]  "Borrador - IMPLEMENTACIÓN DE LA METODOLOGÍA DE LAS 5'S PARApdf" fechado del 2022-09-02
0.2% 7 resultados

- ✓ [71]  aprendeia.com/category/machine-learning-o-aprendizaje-automatico/page/5/
0.4% 10 resultados

- ✓ [72]  keepcoding.io/blog/redes-neuronales-convolucionales/
0.4% 9 resultados

- ✓ [73]  core.ac.uk/download/pdf/41780722.pdf
0.3% 7 resultados

- ✓ [74]  repositorio.upse.edu.ec/bitstream/46000/2374/1/UPSE-TET-2015-0007.pdf
0.2% 6 resultados

- ✓ [75]  blog.pangeanic.com/es/valor-f-asegurando-calidad-de-la-ta-y-anonimizacion
0.4% 6 resultados

- ✓ [76]  hmong.es/wiki/U-Net
0.3% 3 resultados

- ✓ [77]  www.juanbarrios.com/redes-neurales-convolucionales/
0.3% 7 resultados

- ✓ [78]  vincentblog.xyz/posts/conceptos-importantes-en-machine-learning
0.3% 8 resultados

- ✓ [79]  aprendeia.com/que-es-deep-learning/
0.2% 5 resultados

- ✓ [80] es.wikipedia.org/wiki/Red_neuronal_convolucional
0.3% 6 resultados

- ✓ [81] medium.com/neuron4/fine-tuning-en-reconocimiento-de-imagenes-mediante-deep-learning-c656ae728d73
0.2% 7 resultados

- ✓ [82] scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1684-18592013000100010
0.2% 6 resultados

- ✓ [83] www.redalyc.org/pdf/869/86926976002.pdf
0.2% 4 resultados

- ✓ [84] es.linkedin.com/learning/como-obtener-valor-con-machine-learning-en-tu-empresa/aprendizaje-no-supervisado
0.2% 8 resultados

- ✓ [85] randomaccessnoticias.com/inteligencia-artificial/aplicaciones-y-tecnicas-de-ml-ai-comprender-las-tecnicas-y-de-blackyellow-febrero-20
0.2% 4 resultados

- ✓ [86] repositorio.inta.gov.ar/xmlui/bitstream/handle/20.500.12123/13298/INTA_CRBsAsNorte_EEASanPedro_Aballay_MM_et-al_Aplicación
0.3% 3 resultados

- ✓ [87] es.wikipedia.org/wiki/Acuicultura
0.2% 3 resultados

- ✓ [88] medium.com/@torenunez/detrás-de-la-magia-del-aprendizaje-automático-machine-learning-18c528f1901d
0.2% 5 resultados

- ✓ [89] oa.upm.es/69305/1/TFG_SARA_SANZ_GUIRADO.pdf
0.1% 5 resultados

- ✓ [90] tesis.ipn.mx/bitstream/handle/123456789/1001/2520_2007_ESIQIE_MAESTRIA_delossantos_ramos_williamns.pdf?sequence=1&isAll
0.1% 3 resultados

- ✓ [91] www.i-ciencias.com/pregunta/183278/como-convertir-una-capa-totalmente-conectada-en-una-capa-convolucional
0.1% 3 resultados

- ✓ [92] hmong.es/wiki/Lloyd_A._Jeffress
0.1% 5 resultados

- ✓ [93] www.iebschool.com/blog/redes-neuronales-convolucionales-big-data/
0.2% 4 resultados

- ✓ [94] ["Informe de Trabajo de Suficiencia Profesional - VFINAL \(2\).pdf" fechado del 2023-03-14](#)
0.1% 4 resultados

- ✓ [95] learn.microsoft.com/es-es/archive/msdn-magazine/2013/december/test-run-radial-basis-function-network-training
0.1% 3 resultados

- ✓ [96] [repository.upb.edu.co/bitstream/handle/20.500.11912/3543/Operaciones de detección y clasificación de señales en imágenes.pdf?seq](https://repository.upb.edu.co/bitstream/handle/20.500.11912/3543/Operaciones%20de%20detección%20y%20clasificación%20de%20señales%20en%20imágenes.pdf?seq)
0.1% 3 resultados

- ✓ [97] www.scielo.cl/scielo.php?pid=S0718-50732020000200111&script=sci_arttext
0.1% 3 resultados

- ✓ [98] www.virtualpro.co/biblioteca/una-red-totalmente-convolucional-modificada-para-la-identificacion-de-grietas-en-comparacion-con-los-me
0.1% 4 resultados

- ✓ [99] www.mathworks.com/help/deeplearning/getting-started-with-deep-learning-toolbox_es.html
0.1% 3 resultados

- ✓ [100] 1library.co/document/q7w962nz-implementacion-prototipo-hexapodo-teledirigido-contribucion-limpieza-aplicado-tecnologia.html
0.1% 3 resultados

- ✓ [101] learn.microsoft.com/es-es/dynamics365/finance/finance-insights/confusion-matrix
0.1% 4 resultados

52 páginas, 10064 palabras

Nivel del plagio: 21.9% seleccionado / 22.6% en total

293 resultados de 102 fuentes, de ellos 99 fuentes son en línea.

Configuración

Directiva de data: *Comparar con fuentes de internet, Comparar con documentos propios*

Sensibilidad: *Alta*

Bibliografía: *Excluir texto*

Detección de citas: *Reducir PlagLevel*

Lista blanca: 2 - 137654315; 135205535

Dedicatória

Dedico este trabajo a Dios por prestarme salud, vida e inteligencia para continuar siendo un buen profesional en este mundo.

A mis Padres por siempre apoyarme y siempre darme sus consejos que aportaron en mi desarrollo profesional y humano, también por confiar en mí y en mis capacidades.

A la Universidad La Salle por permitir llenarme de conocimientos teóricos y prácticos, en especial a mis maestros quienes a los cuales considero como amigos, gracias por su esfuerzo y vocación impartieron su conocimiento el cual usaré en cada uno de mis actos a lo largo de mi vida profesional.

Resumen

En esta tesis se examina la detección y segmentación de peces con el objetivo de mejorar la crianza de truchas. La crianza de truchas enfrenta desafíos en la identificación y seguimiento preciso de los peces, lo que dificulta el monitoreo de su crecimiento y salud, ayudando a determinar la abundancia relativa y rastrear los cambios en la población de peces.

Para lograr el objetivo, se analizan los modernos algoritmos de detección de objetos basados en deep learning. A partir de este análisis, se han identificado las técnicas que se ajustan mejor a esta investigación, siendo esta YOLOv5 para la detección y U-Net para la segmentación.

Dado que para el entrenamiento es necesario una gran cantidad de datos, se utilizaron distintos Datasets públicos y un dataset generado específicamente para esta investigación. Estos conjuntos de datos proporcionaron la diversidad necesaria para entrenar y evaluar los modelos de detección y segmentación de peces de manera más precisa y confiable.

También se presenta una herramienta que permitirá usar los modelos presentados desde una terminal, haciendo que su uso sea práctico y sencillo, facilitando así su implementación en entornos de crianza de truchas.

En conclusión, esta investigación ha demostrado que el uso de los algoritmos de detección YOLOv5 y de segmentación U-Net, junto con la utilización de diversos conjuntos de datos, ha permitido mejorar significativamente la precisión y el rendimiento en la detección y segmentación de peces. Estos avances pueden tener un impacto positivo en la crianza de truchas, al facilitar el monitoreo y seguimiento de los peces de manera más eficiente y confiable.

Abstract

This thesis examines the detection and segmentation of fish with the aim of improving trout farming. Trout farming faces challenges in accurately identifying and tracking fish, which hinders the monitoring of their growth and health, helping determine relative abundance and track changes in the fish population.

To achieve the objective, modern object detection algorithms based on deep learning are analyzed. From this analysis, the techniques that best fit this research have been identified, namely YOLOv5 for detection and U-Net for segmentation.

Since a large amount of data is required for training, various public datasets and a dataset specifically generated for this research were used. These datasets provided the necessary diversity to train and evaluate fish detection and segmentation models more accurately and reliably.

A tool is also presented that will allow the use of the models presented from a terminal, making their usage practical and straightforward, thereby facilitating their implementation in trout farming environments.

In conclusion, this research has demonstrated that the use of YOLOv5 detection and U-Net segmentation algorithms, along with the utilization of diverse datasets, has significantly improved the accuracy and performance in fish detection and segmentation. These advancements can have a positive impact on trout farming by facilitating more efficient and reliable monitoring and tracking of fish.

Palabras clave

Detección de peces

Segmentación de peces

Crianza de truchas

Algoritmos de aprendizaje profundo

YOLOv5

U-Net

Precisión y recall

Monitoreo de peces

Seguimiento de población

Acuicultura

Herramienta basada en terminal

Implementación en criaderos de peces

Evaluación de rendimiento

Índice general

Palabras clave	v
Lista de abreviaturas	viii
Índice de figuras	ix
Índice de cuadros	x
1 Introducción	1
1.1 Motivación	1
1.2 Objetivos	1
1.2.1 Objetivo General	1
1.2.2 Objetivos Específicos	1
1.3 Descripción del problema	2
1.4 Contribuciones	2
1.5 Organización del Trabajo	3
2 Marco Conceptual	4
2.1 Machine learning	4
2.1.1 Algoritmos de aprendizaje automático	4
2.1.2 Tipos de aprendizaje	5
2.1.3 Redes neuronales	6
2.1.4 Métricas de evaluación de Algoritmos	7
2.2 Deep learning	8
2.2.1 Deep Neural Networks	9
2.2.2 Convolutional Neural Network	9
2.2.3 Detección de objetos	10
2.2.4 Segmentación de objetos	12
3 Estado del Arte	15
3.1 Detección de peces	15
3.2 Segmentación de peces	16
3.3 Conteo de peces	17
4 Propuesta	19
4.1 Método	19

4.1.1	Herramientas	19
4.2	Entrenamiento	19
4.2.1	Confección de dataset	20
4.2.2	Detección	20
4.2.3	Segmentación	21
4.2.4	Conteo	21
4.3	Creación de Herramienta	21
5	Resultados	25
5.1	Preparación de base de datos	25
5.2	Resultados de desempeño	28
5.3	Discusión	30
6	Conclusiones	34
	Bibliografía	35

Lista de abreviaturas

API	Application Programming Interface
AI	Inteligencia Artificial
AP	Precisión media
CNN	Red neuronal convolucional
CPU	Unidad Central de Procesamiento
DVV	Red neuronal profunda
FN	Falso negativo
FP	Falso positivo
GPU	Unidad de procesamiento gráfico
IDE	Entorno de Desarrollo Integrado
MAP	Media Precisión media
VP	Verdadero positivo
VN	Verdadero negativo
YOLO	You only look once

Índice de figuras

2.1	Arquitectura de una CNN	10
4.1	El modelo propuesto con una estructura simplificada de cada etapa.	23
4.2	Red Neuronal U-Net utilizada para la segmentación	24
5.1	Ejemplo de Deteccion-1 DB. Peces fuera del mar en superficies solidas	26
5.2	Ejemplo de Deteccion-2 DB. Peces dentro y fuera del agua incluidas laminas de peces	27
5.3	Ejemplo de VeoxDB. Gran cantidad de peces por debajo del agua	27
5.4	Ejemplo de cajas usadas para detección. Enmarcan al pez	28
5.5	Ejemplo de base de datos segmentación. Imagenes de peces fuera del agua frente a un fondo solido	29
5.6	Ejemplo de base de datos segmentación. Se encuentra imagen base y mascarará para la segmentación	30
5.7	Ejemplo de base de datos conteo. Imagenes de peces bajo del mar	31
5.8	Curva de precision de la base de datos Deteccion-1 DB	31
5.9	Curva de precision de la base de datos Deteccion-2 DB	32
5.10	Comparación de curva de precision de la base de datos VeoxDB v1 y v2	32
5.11	Curva de precisión de la base de datos de Segmentación	33

Índice de cuadros

2.1	Matriz de confusión	7
4.1	Parámetros usados para la detección con Yolo	20
4.2	Parámetros usados para el conteo con DeepFish	21
5.1	Resultados modelo YOLOv5	28
5.2	Resultados modelo U-Net	29
5.3	Resultados modelo DeepFish	30

Capítulo 1

Introducción

1.1. Motivación

Hoy en día las tecnologías de inteligencia artificial son ampliamente utilizadas en distintos ámbitos actuales. En la acuicultura estas tecnologías de inteligencia artificial pueden mejorar en gran medida la productividad pesquera [1], algunos de estos métodos de aprendizaje profundo son usados con el modelo YOLO para extraer el número y las trayectorias de comportamiento de los peces a través de cámaras subacuáticas o para realizar la segmentación detectando la cabeza y cola de cada pez.

En la crianza de truchas hay muchas tareas como son la medición, el marcado de peces, el peso, la biomasa, etc. que diversos criaderos realizan de forma empírica llevando muchas veces una mala técnica provocando una gran mortalidad. Estos productores usan en su mayoría granjas artesanales [2]. Sin duda la crianza de truchas es una tarea que ayuda al desarrollo de muchas comunidades, pero sin un buen cuidado en la crianza de peces puede haber un gran porcentaje de mortandad generando muchas pérdidas [3].

Este proyecto busca proponer un nuevo método sobre los métodos usados actualmente por los criaderos de truchas que utiliza las tecnologías de inteligencia artificial para generar herramientas que no sean intrusivas para los peces disminuyendo en gran medida la mortalidad de estos.

1.2. Objetivos

1.2.1. Objetivo General

Proponer una herramienta para la detección y conteo de peces a partir de secuencias de vídeo utilizando deep learning.

1.2.2. Objetivos Específicos

- (a) Generar un dataset completo y representativo de imágenes de peces;
- (b) Evaluar diversas técnicas de detección y segmentación de peces basadas en algoritmos de aprendizaje profundo;

- (c) Desarrollar un modelo propuesto que integre las técnicas de detección, segmentación y conteo de peces en videos;
- (d) Desarrollar herramienta práctica y de fácil uso que permita la detección, segmentación y conteo automatizado de peces en entornos de crianza;

1.3. Descripción del problema

El pez es una fuente de alimentación principal en todo el mundo, así como una parte esencial de los ecosistemas marinos. El pez ocupa una dieta importante en más de 3 millones de habitantes del mundo [4]. En Perú, según la Organización de las Naciones Unidas para la Alimentación y la Agricultura (FAO), en 2006 ocupó el quinto puesto en producción pesquera, y en el primer semestre de 2018, Perú exportó 839,3 toneladas de peces.

El sistema de cría de peces es muy diverso. Puede llevarse a cabo con agua de mar, tanto en tierra como en alta mar, agua dulce o una mezcla de ambas. Los peces pueden ser cultivados directamente en agua (sistemas abiertos) o en instalaciones con condiciones totalmente controladas (sistemas cerrados). Sin embargo, las técnicas actuales utilizadas para el conteo, detección y segmentación de peces presentan varios problemas.

En muchos criaderos, estas técnicas se realizan de forma manual mediante muestreo y fórmulas estadísticas, lo cual proporciona la información necesaria para la crianza, como raciones alimentarias, población alimentaria e incluso la variación del tamaño entre los especímenes. Sin embargo, estos métodos causan estrés a ciertos especímenes, lo que en muchos casos provoca su muerte.

Además, las técnicas manuales de conteo y seguimiento de peces son laboriosas, consumen mucho tiempo y son propensas a errores humanos. Esto dificulta el monitoreo constante y preciso de la población de peces, así como el seguimiento de su crecimiento y salud.

Para solucionar estos problemas, se propone desarrollar un sistema que, a partir de una entrada de vídeo, pueda detectar, segmentar y contar los peces en tiempo real mientras se procesa el video. Esta solución permitirá un monitoreo constante y automatizado, evitando el estrés y la mortalidad innecesaria de los peces, y brindando información precisa sobre la abundancia relativa y los cambios en la población de peces.

En conclusión, los desafíos asociados con el conteo, detección y segmentación de peces en entornos de cría representan una necesidad urgente de desarrollar soluciones automatizadas y precisas. Mediante la implementación de un sistema que aborde estos problemas, se espera mejorar significativamente el monitoreo y seguimiento de los peces, así como la eficiencia y la salud de la industria acuícola.

1.4. Contribuciones

Las principales contribuciones de este trabajo son describas a seguir:

- (a) Una herramienta que permita la detección y conteo de peces;
- (b) Publicación del dataset creado para esta publicación;

1.5. Organización del Trabajo

En el [Capítulo 2](#) abordamos el marco teórico necesario para entender el desarrollo de esta tesis. En el [Capítulo 3](#) se observa y analiza los trabajos relacionados más importantes desarrollados en los últimos años, se muestra cada uno de los métodos analizados. En el [Capítulo 4](#) se expone la propuesta de este trabajo de investigación. En el [Capítulo 5](#) se presentan los experimentos y resultados de validación para este trabajo de investigación. Finalmente, en el [Capítulo 6](#) son expuestas las conclusiones del presente trabajo así como también las direcciones para continuar con el mismo en la sección de trabajos futuros.

Capítulo 2

Marco Conceptual

En este capítulo se presentarán conceptos necesarios para el correcto entendimiento de esta tesis abarcan las tecnologías utilizadas de visión artificial.

2.1. Machine learning

El machine learning es una rama de la inteligencia artificial que se enfoca en el diseño y desarrollo de algoritmos capaces de aprender de los datos y realizar predicciones sobre ellos. El objetivo principal de esta técnica es permitir que las computadoras mejoren automáticamente su desempeño en una tarea específica a través de la experiencia [5]. A menudo, el aprendizaje automático se relaciona con otros campos como la estadística, la minería de datos y el aprendizaje profundo.

Existen varios tipos de algoritmos de aprendizaje automático, incluyendo el aprendizaje supervisado, el aprendizaje no supervisado, el aprendizaje de refuerzo y el aprendizaje semi-supervisado. Los dos tipos de tareas más comunes en el aprendizaje automático son la clasificación y la regresión [6].

La clasificación se refiere a la tarea de asignar una etiqueta a una entrada determinada, como etiquetar una imagen de un gato como 'gato'. Por otro lado, la regresión se trata de predecir un valor continuo, como el precio de una acción basado en datos históricos [6].

La precisión es una de las principales métricas para evaluar la calidad de un modelo de aprendizaje automático. Esta medida representa el porcentaje de predicciones correctas realizadas por el modelo. Además, otras métricas de evaluación comunes incluyen la precisión, el recall y el f1-score [5].

2.1.1. Algoritmos de aprendizaje automático

Los algoritmos de aprendizaje automático son un conjunto de herramientas que permiten a las computadoras aprender de los datos de manera autónoma. Estos algoritmos ajustan automáticamente sus parámetros para mejorar el rendimiento de un sistema de aprendizaje automático, por lo que hay una gran variedad de algoritmos con diferentes fortalezas y debilidades. El mejor algoritmo para un problema concreto depende de la naturaleza de los datos y del resultado deseado [7].

A continuación, se presentan algunos de los algoritmos de aprendizaje automático más comunes: [8]

- **Regresión lineal:** Es un algoritmo de aprendizaje supervisado que predice una variable de resultado continua (y) a partir de un conjunto de variables predictoras (x). La regresión lineal es una poderosa herramienta para comprender las relaciones entre variables y para hacer predicciones sobre eventos futuros.
- **Regresión logística:** Es un algoritmo de aprendizaje supervisado que predice una variable de resultado binaria (y) a partir de un conjunto de variables predictoras (x). La regresión logística también es una herramienta poderosa para comprender las relaciones entre variables y hacer predicciones sobre eventos futuros.
- **Árbol de decisión:** Es un algoritmo de aprendizaje supervisado no lineal que predice una variable de resultado categórica (y) a partir de un conjunto de variables predictoras (x). Los árboles de decisión son una herramienta poderosa para entender las relaciones entre variables y hacer predicciones sobre eventos futuros.
- **Random Forest:** Es un algoritmo de aprendizaje conjunto que predice variables de resultado continuas o categóricas (y) a partir de un conjunto de variables predictoras (x). Los bosques aleatorios son una herramienta poderosa para comprender las relaciones entre variables y hacer predicciones sobre eventos futuros.
- **Árboles de gradiente reforzado:** Es un algoritmo de aprendizaje conjunto que predice variables de resultado continuas o categóricas (y) a partir de un conjunto de variables predictoras (x). Los árboles de potencia de gradiente son una herramienta poderosa para entender las relaciones entre variables y hacer predicciones sobre eventos futuros.

|

2.1.2. Tipos de aprendizaje

Los tipos de aprendizaje automático se dividen en 3 tipos: [9]

Aprendizaje supervisado. Se entrenan utilizando datos etiquetados, donde una etiqueta indica la salida correcta para una entrada determinada. Los algoritmos aprenden un mapeo de la entrada a la salida basándose en los datos de entrenamiento, y luego pueden hacer predicciones para nuevas entradas.

Los algoritmos de aprendizaje supervisado pueden utilizarse para la regresión, o predicción de valores continuos, y la clasificación, o predicción de etiquetas de clases discretas.

Aprendizaje no supervisado. Se entrenan con datos no etiquetados. Los algoritmos aprenden a encontrar patrones y estructuras en los datos, pero no requieren un resultado específico. El aprendizaje no supervisado puede utilizarse para tareas como la agrupación de puntos de datos en grupos similares y la reducción de la dimensionalidad, es decir, la reducción del número de características de los datos.

Aprendizaje por refuerzo. Se entrenan utilizando una señal de retroalimentación, o recompensa, que indica el rendimiento del algoritmo. Los algoritmos aprenden a realizar acciones que maximizan la recompensa esperada y pueden utilizarse para tareas como los juegos o el control de agentes robóticos.

2.1.3. Redes neuronales

Una red neuronal es un algoritmo de aprendizaje automático que se utiliza para modelar patrones complejos en los datos. Las redes neuronales son similares a otros algoritmos de aprendizaje automático, pero están compuestas por un gran número de nodos de procesamiento interconectados, o neuronas, que pueden aprender a reconocer patrones de datos de entrada [10].

Las redes neuronales se utilizan para diversas tareas, como el reconocimiento de patrones, la clasificación y la predicción. Se han utilizado para modelar sistemas complejos como el cerebro humano y se han aplicado a tareas como el reconocimiento de imágenes, el reconocimiento del habla y el procesamiento del lenguaje natural [10].

Perceptrón

Un perceptrón es un tipo de red neuronal artificial que se utiliza en el aprendizaje supervisado para clasificar entradas. Es una red neuronal de una sola capa que puede clasificar datos lineales separables. El perceptrón es un elemento básico de las redes neuronales. El algoritmo del perceptrón fue desarrollado por Frank Rosenblatt en la década de 1950. Rosenblatt se inspiró en los trabajos del neurofisiólogo Warren McCulloch y del matemático Walter Pitts, que demostraron que las neuronas del cerebro podían realizar operaciones lógicas sencillas [11].

El perceptrón es un modelo sencillo pero potente de una neurona. Puede utilizarse para resolver problemas de clasificación lineal, como determinar si una entrada pertenece a una determinada clase o no. El algoritmo del perceptrón funciona ajustando los pesos de las entradas de la neurona, de modo que ésta se dispara cuando recibe una entrada que es miembro de la clase, y no se dispara cuando recibe una entrada que no es miembro de la clase [12].

El perceptrón se puede entrenar con un conjunto de datos proporcionándole un conjunto de entradas y sus correspondientes salidas. El perceptrón ajustará sus pesos para poder clasificar correctamente las entradas.

El perceptrón es una poderosa herramienta de aprendizaje supervisado, pero está limitado por su capacidad de aprender sólo datos lineales separables. Esto significa que el perceptrón no puede aprender a clasificar datos que no son linealmente separables, como los datos representados por una función no lineal [11].

2.1.4. Métricas de evaluación de Algoritmos

Existen varias métricas de evaluación para los algoritmos de aprendizaje automático, y la elección de la métrica depende de la tarea específica para la que se utiliza el algoritmo. Las métricas usadas en esta investigación son: [5]

Matriz de confusión. Es un cuadro en el que se registra el número de predicciones realizadas de acuerdo a la categoría encajada. Cada clase a predecir es un conjunto de datos binarios que puede ser dos posibles valores: clase positiva y clase negativa, en la Tabla 1 se ve esto. El número de predicciones positivas y negativas que el modelo predice correctamente se denomina True Positive (TP) y True Negative (TN) respectivamente. Las predicciones mal clasificadas se conocen como False Positive (FP) y False Negative (FN).

Cuadro 2.1: *Matriz de confusión*

	Prediction	
	Positive	Negative
True	TP	FN
False	FP	TN

Fuente: Propia

Accuracy. Es la proporción de predicciones correctas realizadas por el algoritmo entre todas las predicciones realizadas.

$$accuracy = \frac{\text{predicciones correctas}}{\text{total predicciones}} \quad (2.1)$$

Precision. Es la proporción de predicciones correctas realizadas por el algoritmo de todas las predicciones que fueron positivas.

$$precision = \frac{TP}{TP + FP} \quad (2.2)$$

Recall. La recuperación es la proporción de predicciones correctas realizadas por el algoritmo entre todos los casos positivos reales.

$$recall = \frac{TP}{TP + FN} \quad (2.3)$$

F1 score. La puntuación F1 es una medida de precisión que tiene en cuenta tanto la precisión como la recuperación.

$$F1score = 2 * \frac{precision * recall}{precision + recall} \quad (2.4)$$

Error Absoluto Medio (MAE): Es la media de las diferencias absolutas entre las predicciones del modelo y los valores reales. Se calcula de la siguiente manera:

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (2.5)$$

Donde:

- n es el número total de muestras en el conjunto de datos.
- y_i es el valor real (etiqueta) de la muestra i .
- \hat{y}_i es el valor predicho por el modelo para la muestra i .

Error medio absoluto en Cuadrícula (GAME): Es una medida utilizada para evaluar el desempeño de algoritmos de detección de objetos en tareas de detección y localización. Proporciona una medida de la precisión en la localización de objetos alrededor de su posición real en una imagen. Se calcula de la siguiente manera:

$$GAME = \frac{1}{N} \sum_{i=1}^N \left(\frac{1}{A_i} \sum_{j=1}^M d(c_i, \hat{c}_j) \right) - \hat{y}_i)^2 \quad (2.6)$$

Donde:

- N es el número total de objetos de referencia.
- M es el número total de objetos detectados.
- A_i es el área del objeto de referencia i .
- c_i es el centro del objeto de referencia i .
- \hat{c}_j es el centro del objeto detectado j .
- $d(c_i, \hat{c}_j)$ es la distancia euclidiana entre los centros de los objetos de referencia y detectados.

Pérdida (Loss): Es una medida de la discrepancia entre las predicciones del modelo y los valores reales. La pérdida se utiliza para optimizar el modelo durante el entrenamiento. Se calcula de la siguiente manera:

$$Loss = -\frac{1}{N} \sum_{i=1}^N \sum_{j=1}^C y_{ij} \log(\hat{y}_{ij}) \quad (2.7)$$

Donde:

- N es el número total de muestras en el conjunto de datos.
- C es el número de clases.
- y_{ij} es el valor verdadero (etiqueta) de la clase j para la muestra i .
- \hat{y}_{ij} es el valor predicho por el modelo para la clase j en la muestra i .

2.2. Deep learning

Deep learning es un subconjunto del aprendizaje automático que se ocupa de los algoritmos inspirados en la estructura y el funcionamiento del cerebro llamados redes neuronales artificiales. Las redes neuronales son un conjunto de algoritmos diseñados para reconocer patrones. Interpretan los datos sensoriales a través de una especie de percepción mecánica, etiquetando o agrupando la entrada bruta. Los patrones que reconocen son numéricos, contenidos en vectores, a los que deben traducirse todos los datos del mundo real, ya sean imágenes, sonidos, textos o series temporales.

Deep learning es una nueva área de investigación del machine learning, que se ha introducido con el objetivo de acercar el machine learning a uno de sus objetivos originales: la inteligencia artificial. Deep learning consiste en el aprendizaje de múltiples capas de representación y abstracción que

ayudan a dar sentido a datos como las imágenes, el sonido y el texto [11].

2.2.1. Deep Neural Networks

Los Deep Neural Networks (DNN) son potentes modelos de aprendizaje automático que han demostrado su éxito en una amplia gama de tareas, como la clasificación de imágenes, el procesamiento del lenguaje natural y el aprendizaje por refuerzo. Están compuestas por múltiples capas de neuronas artificiales, y cada capa es capaz de aprender una representación de los datos de entrada. Las representaciones aprendidas por las diferentes capas de una DNN pueden utilizarse para diferentes tareas, como la clasificación, la regresión y la predicción [13].

Las DNN son especialmente adecuadas para tareas que requieren el aprendizaje de representaciones complejas. Por ejemplo, se han utilizado para aprender representaciones de imágenes que pueden utilizarse para la clasificación de objetos. Las DNN también se han utilizado para aprender representaciones de texto que pueden utilizarse para el análisis de sentimientos. Además, se han utilizado para aprender representaciones de datos que pueden utilizarse para el modelado predictivo [14].

Las DNN también son adecuadas para tareas que requieren el aprendizaje de grandes cantidades de datos. Por ejemplo, se han utilizado para aprender de grandes conjuntos de datos compuestos por millones de imágenes. Las DNN también se han utilizado para aprender de grandes conjuntos de datos que consisten en millones de documentos de texto. Además, las DNN se han utilizado para aprender de grandes conjuntos de datos que consisten en millones de puntos de datos [14].

Las DNN se componen de una capa de entrada, capas ocultas y una capa de salida. Las capas ocultas de una DNN pueden estar compuestas por diferentes tipos de neuronas, como las neuronas convolucionales, las neuronas de agrupación y las neuronas totalmente conectadas.

2.2.2. Convolutional Neural Network

Una red neuronal convolucional (CNN) es un tipo de red neuronal de aprendizaje profundo que se utiliza generalmente para procesar datos visuales. Las CNN son similares a las redes neuronales ordinarias en el sentido de que están compuestas por neuronas que tienen pesos y sesgos aprendibles. Sin embargo, las CNN utilizan una arquitectura especial que se adapta particularmente bien a las imágenes [13].

Las CNN están compuestas por una serie de capas, cada una de las cuales realiza una tarea específica. La primera capa suele ser una capa convolucional que realiza convoluciones sobre la imagen de entrada (de ahí el nombre de 'red neuronal convolucional'). La salida de la capa convolucional suele ser un mapa de características, que es un tensor tridimensional (altura, anchura, profundidad). La siguiente capa suele ser una capa de agrupación que reduce la muestra del mapa de características. La última capa suele ser una capa totalmente conectada que produce la salida final de la CNN [15].

Las CNN pueden utilizarse para diversas tareas, como la clasificación de imágenes, la detección de objetos y el reconocimiento de caras. Las CNN han tenido mucho éxito en el campo de la clasificación de imágenes, especialmente cuando el conjunto de datos es grande y las imágenes son

ARQUITECTURA DE UNA CNN

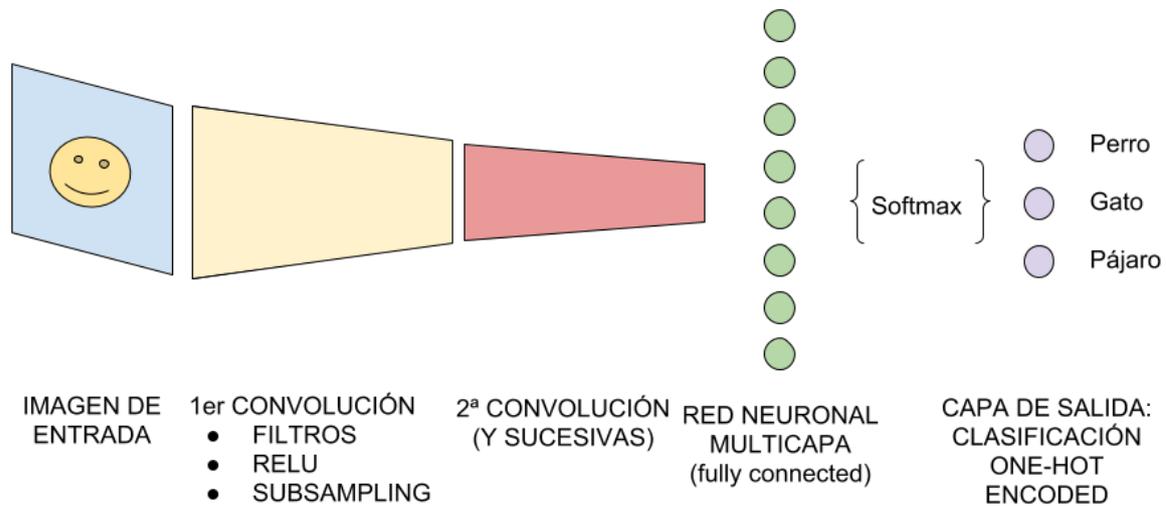


Figura 2.1: *Arquitectura de una CNN*

Fuente: ¿Cómo funcionan las Convolutional Neural Networks? Visión por Ordenador [16]

de alta resolución. Las CNN también son adecuadas para la detección de objetos, ya que pueden aprender a detectar objetos en las imágenes [15].

2.2.3. Detección de objetos

La detección de objetos es una tecnología informática relacionada con la visión por ordenador y el procesamiento de imágenes que se ocupa de detectar instancias de objetos semánticos de una determinada clase (como seres humanos, edificios o coches) en imágenes y vídeos digitales. Entre los dominios bien investigados de la detección de objetos se encuentran la detección de rostros y la detección de peatones. La detección de objetos tiene aplicaciones en muchos ámbitos de la visión por ordenador, como la recuperación de imágenes y la videovigilancia [17].

Los primeros trabajos sobre la detección de objetos se centraron en características elaboradas a mano y en una heurística sencilla para clasificar los objetos. Estos métodos se enfrentaban a la variedad de clases, formas y apariencias de los objetos. Los enfoques más recientes han utilizado el aprendizaje automático para aprender características directamente de las imágenes que luego se utilizan para la detección. Estas características suelen aprenderse entrenando una red neuronal convolucional (CNN) en un gran conjunto de datos de imágenes que contienen los objetos deseados [18].

Un marco común para la detección de objetos es la R-CNN (CNN basada en la región), que primero extrae las regiones de interés de una imagen utilizando una búsqueda selectiva y luego clasifica cada región utilizando una CNN. Otro marco es el Detector de Disparo Único (SSD), que utiliza una CNN para predecir directamente los cuadros delimitadores y las probabilidades de clase de los

objetos de una imagen [18].

Hay muchos otros marcos de detección de objetos que se han propuesto en los últimos años, como el marco You Only Look Once (YOLO), el framework Fast R-CNN y el framework Faster R-CNN [19][20].

La detección de objetos es un problema difícil que se ha estudiado ampliamente en la literatura de visión por ordenador. Hay muchos enfoques diferentes para resolver este problema, y no se garantiza que ningún enfoque funcione mejor para todos los conjuntos de datos y todas las clases de objetos. Sin embargo, los recientes avances en el aprendizaje profundo han llevado a mejoras significativas en el estado del arte para la detección de objetos [17].

El entrenamiento y la comprobación de los modelos de detección de objetos pueden ser largos y costosos, debido a la necesidad de grandes cantidades de datos anotados. Sin embargo, hay muchas herramientas y bibliotecas de código abierto que pueden utilizarse para entrenar y probar modelos de detección de objetos, como la API de detección de objetos de TensorFlow y el marco Darknet [21].

Yolo

You Only Look Once, o YOLO, es un algoritmo de detección de objetos en tiempo real. Fue propuesto por Joseph Redmon, Santosh Divvala, Ross Girshick y Ali Farhadi en el artículo 'You Only Look Once: Unified, Real-Time Object Detection', publicado en mayo de 2016 [22].

El algoritmo se basa en una red neuronal convolucional profunda (CNN) que se entrena con datos con ubicaciones de objetos conocidas. A continuación, la CNN se utiliza para predecir los recuadros delimitadores de los objetos en las nuevas imágenes. El algoritmo YOLO es único porque predice tanto la ubicación como la clase de los objetos en una imagen, sin necesidad de una ventana deslizante o de un paso de propuesta de región [22].

El algoritmo YOLO se ha implementado de diferentes maneras. El documento original de YOLO utilizaba una CNN con 19 capas, mientras que las versiones más recientes han utilizado hasta 152 capas. En el artículo YOLOv5, los autores utilizan una ResNet-50 como base para su red de detección de objetos [23].

El algoritmo YOLO ha demostrado su eficacia en la detección de diversos objetos, como personas, vehículos y animales. También se ha utilizado para aplicaciones como la detección de rostros, la estimación de la postura humana y la detección de textos [22].

El algoritmo YOLO no está exento de limitaciones. Uno de los mayores retos es que requiere muchos datos de entrenamiento para ser eficaz. Otro reto es que YOLO no siempre produce los cuadros delimitadores de mayor calidad.

A pesar de estos problemas, YOLO sigue siendo un algoritmo de detección de objetos robusto

que se utiliza en numerosos sistemas.

Otros métodos de detección

Los algoritmos para la detección de objetos en imágenes con CNN pueden dividirse en dos categorías: los que localizan un objeto en una imagen (es decir, identifican su ubicación), y los que clasifican un objeto (es decir, identifican su etiqueta de clase) [24].

Hay muchos algoritmos diferentes que pueden utilizarse para la detección de objetos en imágenes con las CNN. Algunos algoritmos: [24]

R-CNN(red neuronal convolucional basada en regiones): Este algoritmo primero localiza los objetos en una imagen usando una CNN, y luego clasifica los objetos usando un algoritmo de propuesta de región.

Fast R-CNN: Este algoritmo es similar a la R-CNN, pero utiliza una red de propuesta de regiones (RPN) en lugar de un algoritmo de propuesta de regiones independiente.

Faster R-CNN: Este algoritmo es similar al R-CNN rápido, pero utiliza una capa de agrupación de RoI para mejorar la velocidad y la precisión.

SSD (Single Shot Detector): Este algoritmo también predice directamente los cuadros delimitadores y las probabilidades de clase en una imagen, pero utiliza un enfoque diferente al de YOLO.

2.2.4. Segmentación de objetos

La segmentación de imágenes es el proceso de dividir una imagen en múltiples segmentos. Hay una variedad de formas de realizar la segmentación de imágenes, incluyendo: [25]

Umbralización: El umbral es una forma sencilla de segmentar una imagen convirtiéndola en una imagen binaria, en la que todos los píxeles son blancos o negros. El valor del umbral se suele elegir en función del histograma de la imagen.

Crecimiento de regiones: El crecimiento de la región es un enfoque más sofisticado que comienza con un punto semilla y luego crece la región mediante la adición de píxeles adyacentes que son similares al punto semilla.

Clustering: El clustering es un método para segmentar una imagen en múltiples grupos, o clusters, de píxeles. Los píxeles de cada grupo son similares entre sí, pero diferentes de los píxeles de otros grupos.

Redes neuronales convolucionales: Las CNN son un tipo de algoritmo de aprendizaje profundo especialmente adecuado para la segmentación de imágenes. Las CNN aprenden a segmentar imágenes observando muchos ejemplos de imágenes y sus correspondientes etiquetas.

Existen varias arquitecturas de CNN para la segmentación de imágenes, como FCN, U-Net, Mask R-CNN y SegNet. Cada arquitectura tiene sus propios puntos fuertes y débiles, y la mejor arquitectura para una tarea concreta depende del conjunto de datos y de la aplicación [25].

U-net

U-Net es una red totalmente convolucional para la segmentación de imágenes que fue desarrollada por Olaf Ronneberger et al. en 2015. La red se basa en la red totalmente convolucional y se adapta al problema específico de la segmentación semántica de imágenes [26].

La idea principal de U-Net es utilizar una red totalmente convolucional con una ruta de contracción para codificar una imagen de entrada en una representación de características de baja dimensión, y luego utilizar una ruta de expansión simétrica para decodificar la representación de características de nuevo en una imagen de salida. La ruta de contracción se compone de una serie de capas de muestreo descendente, mientras que la ruta de expansión se compone de una serie de capas de muestreo ascendente. Las capas de muestreo descendente suelen ser capas convolucionales, mientras que las capas de muestreo ascendente suelen ser capas convolucionales transpuestas [26].

Para entrenar un modelo U-Net, se necesita un conjunto de imágenes de entrenamiento y sus correspondientes máscaras de segmentación de verdad. A continuación, el modelo se entrena utilizando un enfoque de aprendizaje supervisado estándar, en el que la pérdida se define normalmente como la entropía cruzada entre las máscaras de segmentación predichas y las máscaras de la verdad sobre el terreno [26].

Una vez entrenado el modelo U-Net, puede utilizarse para segmentar nuevas imágenes. Dada una imagen de entrada, el modelo la codificará primero en una representación de características de baja dimensión. A continuación, el modelo decodificará la representación de características en una imagen de salida. La imagen de salida será una máscara de segmentación del mismo tamaño que la imagen de entrada.

Mask R-CNN

Mask R-CNN es un algoritmo de visión por ordenador que permite segmentar con precisión los objetos de una imagen. Es una extensión del algoritmo de detección de objetos Faster R-CNN. Mask R-CNN es capaz de detectar y segmentar múltiples objetos en una imagen con gran precisión [27].

El algoritmo Mask R-CNN se basa en una red neuronal convolucional profunda. La red se entrena con un gran conjunto de datos de imágenes con máscaras de objetos. La red aprende a generar máscaras de objetos a partir de las imágenes. A continuación, las máscaras se utilizan para segmentar los objetos en nuevas imágenes [27].

Mask R-CNN es un potente algoritmo de visión por ordenador que tiene una amplia gama de

aplicaciones. Puede utilizarse para la detección de objetos, la segmentación de imágenes y el reconocimiento de imágenes.

Capítulo 3

Estado del Arte

3.1. Detección de peces

La detección de peces debajo del agua puede resultar complicada debido a los distintos desafíos que presenta, como la turbiedad, la oscuridad, la deformación de los peces al nadar, entre otros. Sin embargo, existen soluciones tecnológicas para enfrentar estos desafíos.

Una de ellas es el uso de flujo óptico, modelos de mezcla gaussiana y YOLO, un nuevo algoritmo que ha demostrado una precisión del 95,47 % y 91,2 %, mientras que las precisiones normales son del 91,64 % y 79,8 % [28]. Otra solución es el uso de vehículos automáticos bajo el agua (AUV) con visión artificial, que utilizan cámaras binoculares para obtener una mayor cantidad de datos de las imágenes. En este caso, se utiliza YOLO para detectar cajas delimitadoras entre peces y extraer los bordes de los mismos, así como SGBM para predecir el mapa de profundidad y predecir el largo y ancho de cada pez. Gracias a estas tecnologías se pueden obtener resultados precisos en tiempo real [29]. Esta técnica ha demostrado ser eficaz en la detección precisa de peces, aprovechando la información estereoscópica proporcionada por las cámaras binoculares. Sin embargo, su implementación puede requerir recursos adicionales, como la disponibilidad de AUV y cámaras especializadas.

Otra técnica para la detección de peces es la mejora de imágenes con el algoritmo Multi Scale Retinex, la detección mediante YOLO y la identificación de coordenadas para la extracción de características como la cantidad y la trayectoria [30].

También se ha propuesto una técnica que utiliza la información de frames en vídeos de peces para mejorar la detección del pescado. Esta técnica logró una precisión del 93,94 % y 97,06 % [31].

Aunque YOLO parece ser una herramienta eficiente para la detección de peces, se siguen realizando modificaciones para mejorar su rendimiento. Por ejemplo, se aplica en YOLO el cluster de K-means, novel transfer learning y una mejora de la función de pérdida para lograr un rendimiento del 91,30 % [32]. Además, estas técnicas fueron orientadas a una versión antigua de yolo, las mejoras podrían no funcionar correctamente con la versión utilizada en esta tesis.

Para la comercialización de peces, se ha desarrollado un algoritmo de procesamiento de imágenes que clasifica a los peces según la especie y el tamaño. Fue utilizado en carpas (*Cyprinus carpio*),

peces de San Pedro (*Oreochromis* sp.) y salmonetes (*Mugilcephalus*), logrando identificar las especies de peces con una precisión del 100 %, 94 % y 86 %, respectivamente. La estimación de la masa se realiza a través del área de la imagen, lo que resulta en una precisión del 95,4 %, 98,6 % y 98,6 % [33]. Estos algoritmos se basan en el análisis de características morfológicas de los peces, como el área de la imagen. Sin embargo, su aplicación está más orientada a la industria pesquera y puede no ser relevante para la investigación en detección de peces en entornos naturales.

Para la detección de cabeza y cola se ha propuesto una solución basada en el uso de YOLOv4-tiny con Agrupación de Pirámide Espacial (Spatial Pyramid Pooling) [34]. El objetivo de esta propuesta es mejorar el rendimiento de YOLOv4-tiny en la detección precisa de la cabeza y la cola de los peces, utilizando un modelo de bajo coste. Aunque la solución propuesta muestra buenos resultados en la detección de cabeza y cola de peces, su uso en esta investigación puede no ser apropiado debido a diferentes objetivos.

Otra técnica utiliza un vehículo operado a distancia que equipado con una cámara RGB y sensores para detectar patrones de redes buscando fallos asociados a estas. Realizaron pruebas en condiciones reales dando una precisión de 91 % en tareas de reconstrucción de patrones de red y una precisión de 79 % para la detección de daño [35]. Sin embargo, esta técnica puede no ser utilizada en la investigación actual debido a que el enfoque principal está centrado en otros aspectos de detección de peces y no específicamente en la detección de patrones de redes y daños asociados.

Por otro lado presentan un modelo que se ajusta a distintas variaciones del video, como podrían ser variaciones de iluminación, brillo, etc. El sistema presentado consta de tres partes: Aumento de datos en tiempo real, detección de objetos y clasificador de peces. El sistema demuestra un 90 % de precisión para la detección y para la segmentación un 92 % [36].

Proponen un método de detección basado en una red neuronal convolucional al cual se le aplicaron 3 enfoques de optimización: aumento de datos, simplificación de la red y aceleración del proceso de capacitación. Logrando crear un modelo prometedor y con la posibilidad de usarlo para otros objetos marinos [37].

Por último, presentan un enfoque en el que usan redes neuronales y visión artificial logrando detectar peces en videos acústicos. Para esto realizan el tratamiento de las imágenes localizando algunas características de las imágenes para detectar finalmente el pez utilizando YOLOv3. Para la detección de peces este modelo logra un 80 % de detecciones correctas, sin embargo es menos eficiente para la detección de anguilas [38]. Aunque este enfoque podría resultar interesante, en esta investigación no se tratan videos acústicos.

3.2. Segmentación de peces

En el ámbito subacuático, la segmentación de peces puede ser muy útil. Se presentan dos técnicas para la segmentación de la cabeza y la cola de los peces, YOLO y Mask R-CNN. Después de realizar pruebas, se concluye que YOLO tiene una precisión ligeramente mejor con 98.96 %, en comparación con Mask R-CNN con 96.73 % [39].

Se propone una idea interesante para el entrenamiento de una red neuronal, utilizando la información de movimiento de los peces en videos a través de la sustracción de fondo y el flujo óptico, combinando los resultados con la imagen sin procesar para generar regiones candidatas dependientes de los peces. Este enfoque logra una precisión de 87.44 % y 80.02 % [40]. También se presenta una forma de cuantificar las capturas pesqueras mediante el uso de dispositivos electrónicos y una versión modificada de Mask R-CNN para la segmentación de peces, logrando una precisión del 98 % y un recall del 95 % [41], aunque hay una precisión muy buena presenta ciertos problemas con la superposición de peces en las imágenes.

Por otro lado, se plantea una forma de estimar la longitud de los peces, utilizando la segmentación con Mask R-CNN y refinando con gradientes, para luego estimar el tamaño con la capa de profundidad. Este modelo logra una precisión del 99.4 % para peces solos y 98.4 % para peces superpuestos [42]. Además, se utiliza el conjunto de datos DeepFish para crear un sistema de seguimiento automatizado de las capturas de pescado en los mercados de pescado [43]. Aunque los enfoques mencionados presentan logros significativos en la segmentación y estimación de características de los pece es necesario considerar la integración de técnicas adicionales para abordar desafíos más amplios segmentación de peces.

Finalmente, para mejorar los métodos de reconocimiento en entornos subacuáticos, se proponen cuatro etapas: preprocesamiento de video, detección de peces utilizando ResNet, detección mejorada de múltiples peces utilizando la arquitectura RPN y el uso de un método de segmentación de instancia dinámica, demostrando un mejor desempeño en videos subacuáticos [44].

3.3. Conteo de peces

En el campo del conteo de peces, se han propuesto diversos métodos automatizados para estimar la población de salmón del Atlántico y otras especies. Uno de estos métodos utiliza redes neuronales para procesar la información debido a los cambios de forma, angulo y tamaño del pez, y logra una precisión del 95,06 % [45]. Otro método propone detectar automáticamente los peces capturados y medir la longitud de sus cuerpos en los videos de sistemas de seguimiento electronico. Para eso utilizan Mask R-CNN para la detección y la segmentación del pescado, el conteo se realiza a traves de un umbral de distancia y la longitud se estima de a cuerdo a los pixeles extremos mas alejados del pez. El modelo Mask R-CNN alcanzo una precisión de 94.7% y el metodo de conteo alcanzo una precision de 73.37 % [46]. Esta limitación en la precisión del conteo puede afectar la fiabilidad de los resultados y su aplicabilidad en estudios de población

Además, hay un proyecto que busca monitorear peces en riesgo, haciendo uso de dispositivos acústicos, sonar visual y activo, y cámaras ópticas. El dataset planteado fue una sonda de imágenes DIDSON de alta resolución capturados en el río Ocqueoc en Michigan, EE. UU. Estos datos son utilizados con YOLO para la deteccion y clasificación de las 8 especies de peces, posteriormente utiliza el marco de seguimiento de objetos de Norfair para rastrear y contar peces. Demostrando que se puede utilizar la detección, clasificación y rastreo de peces [47] .

En otra técnica se usaron 4 arquitecturas (DenseNet201, InceptionNetV2, Xception y MobileNetV2) para predecir la cantidad de peces y delfines acompañados de funciones de regresión y clasificación. Para la regresión, Densenet201 funcionó mejor para peces y Xception mejor para delfines con errores cuadráticos medios de 2,11 y 0,133 respectivamente. Para la clasificación, InceptionResNetV2 funcionó mejor para peces y MobileNetV2 mejor para delfines con un error medio de 2,07 y 0,245 respectivamente [48]. Aunque se obtuvieron errores cuadráticos medios bajos para la regresión y el error medio para la clasificación, estos enfoques no resuelven directamente el problema del conteo de peces individuales con alta precisión.

Por último, se ha propuesto una técnica que consiste en detectar a los peces frame a frame y utilizar distancia euclidiana entre peces de frames consecutivos siendo que el que tenga menos distancia será el mismo pez [49]. El conteo puede verse afectado por la ocultación o superposición de peces en los frames.

Capítulo 4

Propuesta

En el presente Capítulo se describe el diseño de la propuesta. Cada parte estará dividida en 3 partes: Detección, Segmentación y Conteo.

4.1. Método

Para entrenar y evaluar los modelos de detección y segmentación de peces, se utilizan conjuntos de datos etiquetados manualmente, tanto públicos como generados específicamente para la investigación. Se miden diferentes métricas de rendimiento, como la precisión, el recall y la pérdida, para cuantificar el desempeño de los modelos.

Este enfoque cuantitativo permite obtener resultados objetivos y medibles sobre la eficacia de los modelos en la detección y segmentación de peces, lo que es fundamental para su aplicación práctica en la crianza de truchas y en la gestión de acuicultura en general. Además, permite comparar diferentes técnicas y ajustes de los modelos para determinar las que ofrecen mejores resultados. Junto con esto se requirió una serie de herramientas, las cuales se detallan a continuación.

4.1.1. Herramientas

Google Colab es un Entorno de Desarrollo ofrecido por Google que proporciona un entorno de código como Jupyter Notebook, permitiendo usar Graphic Process Unit (GPU) necesaria para el procesamiento de imágenes. Debido a que los algoritmos de aprendizaje automático/profundo requieren que el sistema tenga una alta velocidad y capacidad de procesamiento (normalmente basada en la GPU) Google Colab es una buena opción, es por eso que en esta investigación es utilizada.

4.2. Entrenamiento

En este apartado se evalúan los entrenamientos realizados para cada una de las etapas de este proyecto.

Se propone un marco de tres etapas, la primera etapa detecta peces dentro de imágenes, para esta etapa se utilizó la técnica YOLO. En la segunda etapa se realiza la segmentación de peces, para esta etapa se utilizó UNet, algoritmo de segmentación usado en ciencias de la computación.

La tercera etapa es la de contar los peces dentro de la imagen, para esta etapa se utilizó DeepFish. La [Figura 4.1](#) ilustra las tres etapas.

4.2.1. Confección de dataset

En esta tesis, se llevó a cabo una colaboración con la empresa VEOX para la recopilación de imágenes en un entorno subacuático. Con el objetivo de obtener imágenes en un entorno real de criado, se realizó una incursión a la región de Puno, donde se encuentran criaderos de truchas.

Durante esta incursión, se utilizaron cámaras subacuáticas para grabar videos en los criaderos de truchas. Estos videos fueron luego procesados para extraer cuadros individuales y obtener así un conjunto de imágenes. El objetivo de esta etapa era capturar la mayor variedad posible de truchas en diferentes poses y condiciones ambientales.

Una vez obtenidas las imágenes, se procedió al etiquetado manual de las mismas. Para este propósito, se utilizó el programa LabelImg, una herramienta de código abierto ampliamente utilizada en la comunidad de visión por computadora para etiquetar objetos en imágenes. El etiquetado consistió en trazar un contorno alrededor de cada trucha presente en la imagen y asignarle una clase correspondiente.

En total, se etiquetaron manualmente 576 imágenes de truchas. El proceso de etiquetado manual permitió generar un conjunto de datos anotados que servirá como base para el entrenamiento y evaluación de los modelos de detección de truchas.

Estas imágenes etiquetadas constituyen una parte fundamental del proceso de entrenamiento de los modelos de detección de truchas, ya que proporcionan la información necesaria para enseñar al modelo a reconocer y localizar las truchas en nuevas imágenes.

4.2.2. Detección

Para la detección de los peces se utilizó Yolov5. Se encontró que Yolov5 presentaba un equilibrio óptimo entre precisión y velocidad de detección, lo que lo hace adecuado para la tarea de detección de peces en ambientes acuáticos. Además, Yolov5 requiere menos tiempo de entrenamiento y es más rápido en la inferencia que otros algoritmos evaluados. Esta técnica permitió tener grandes resultados con respecto a los distintos base de datos probados.

Para la configuración de los parámetros de Yolov5 se realizaron pruebas y experimentos previos para encontrar los valores óptimos que permitieran obtener los mejores resultados en la detección de peces en ambientes acuáticos. La [Tabla 4.1](#) muestra los valores que se encontraron adecuados para las evaluaciones.

Cuadro 4.1: *Parámetros usados para la detección con Yolo*

Parametros	Cantidad
epochs	20
bbox_interval	1
save-period	1

Fuente: Elaboración Propia

epochs: Representa una pasada completa del conjunto de datos de entrenamiento a través del

modelo. `bbox_interval`: El número de epochs que deben transcurrir antes de guardar el modelo.
`save_period`: Se refiere al número de epochs entre cada guardado de modelo.

4.2.3. Segmentación

En cuanto a la segmentación se generó para U-Net una red neuronal como se muestra en [Figura 4.2](#). Los resultados obtenidos mostraron que U-Net logró una alta precisión y sensibilidad en la segmentación de peces, lo que indica que puede detectar de manera efectiva y precisa las regiones de interés en las imágenes de peces. Además, U-Net también presentó una alta especificidad, lo que indica que puede detectar y eliminar eficazmente las regiones no relevantes en la imagen.

4.2.4. Conteo

Para el conteo de peces se utilizó DeepFish. Deepfish es una herramienta de aprendizaje profundo para contar peces en imágenes submarinas. La tecnología DeepFish, al utilizar redes neuronales convolucionales entrenadas para la detección y el conteo de peces, puede proporcionar una solución más eficiente y precisa para el conteo de peces en la acuicultura. La tecnología es capaz de procesar grandes cantidades de imágenes de los peces en tiempo real y proporcionar un recuento preciso y confiable de los peces en los tanques.

La configuración que se maneja para este entrenamiento se ve en la [Tabla 4.2](#)

`batch_size`: Indica el número de entrenamientos que se procesan en una sola iteración del algoritmo de entrenamiento.

`max_epoch`: Indica el número máximo de épocas que se utilizarán para entrenar el modelo.

`model`: Es el tipo de modelo que se está utilizando.

`task`: Es la tarea que se está realizando.

`transform`: Es una técnica de preprocesamiento.

`wrapper`: Es un componente del modelo que envuelve o encapsula la salida del modelo y la adapta para un propósito específico.

Cuadro 4.2: *Parámetros usados para el conteo con DeepFish*

Parametros	Cantidad
<code>batch_size</code>	1
<code>max_epoch</code>	1000
<code>model</code>	inception
<code>task</code>	cont
<code>transform</code>	resize_normalize
<code>wrapper</code>	clf_wrapper

Fuente: Elaboración Propia

4.3. Creación de Herramienta

La fase final del proyecto consiste en crear una herramienta utilizando Python 3.10.1. Para la implementación de los modelos y el procesamiento de las imágenes, se utilizaron las siguientes bibliotecas utilizados para la creación de la herramienta:

- **Python 3.10.1:** Se utilizó Python como lenguaje de programación principal para la implementación de la herramienta.
- **NumPy:** NumPy es una biblioteca fundamental para la computación científica en Python. Se utilizó para realizar operaciones matemáticas y manipulación de matrices.
- **TensorFlow:** TensorFlow es una biblioteca de aprendizaje automático de código abierto ampliamente utilizada. Se utilizó para cargar el modelo UNet en formato .h5 y realizar la segmentación de imágenes.
- **OpenCV:** OpenCV se utilizó para cargar y manipular los frames de los videos, así como para aplicar transformaciones y detecciones en la tarea de detección.
- **ONNX Runtime:** ONNX Runtime se utilizó para cargar y ejecutar los modelos de detección en formato ONNX.
- **DeepFish:** DeepFish es una herramienta especializada en el conteo de objetos en imágenes y videos. Se utilizó para realizar el conteo de truchas en los criaderos.

En la tarea de segmentación, se utilizó un modelo UNet entrenado previamente, el cual se encuentra almacenado en un archivo .h5. TensorFlow se encargó de cargar este modelo y realizar la segmentación de las imágenes utilizando sus capacidades de inferencia.

En la tarea de detección, se utilizó OpenCV para cargar el modelo en formato ONNX, que pudo haber sido entrenado con alguna arquitectura como YOLO o SSD. OpenCV permitió cargar y ejecutar este modelo para detectar objetos de interés en cada imagen del video.

En la tarea de conteo, se utilizó la herramienta DeepFish, la cual es especialmente diseñada para contar objetos en imágenes y videos. Se configuró y utilizó para realizar el conteo de truchas en los criaderos.

Para la evaluación del algoritmo a través de la consola, se debe utilizar el siguiente comando:

```
python evaluation.py -md TASK -inputDATAVIDEO
```

Siendo las variables $\{\dots\}$:

1. TASK: det, seg, cont.
2. DATAVIDEO: Ruta del video a ser tratada.

Al ejecutar este comando, la herramienta cargará el modelo correspondiente y realizará la evaluación en cada imagen del video, generando los resultados esperados según la tarea seleccionada.

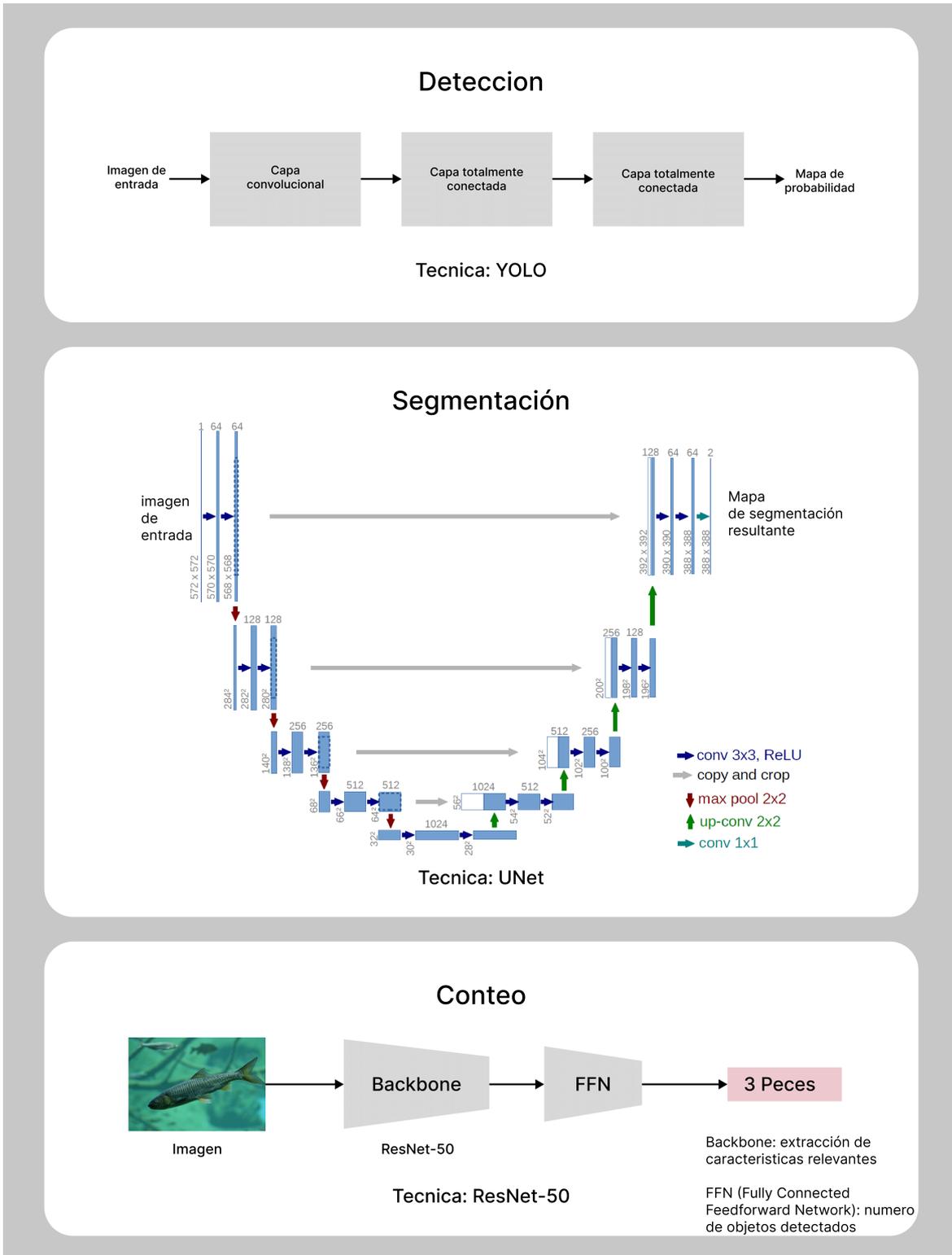


Figura 4.1: El modelo propuesto con una estructura simplificada de cada etapa.

Fuente: [26][50].

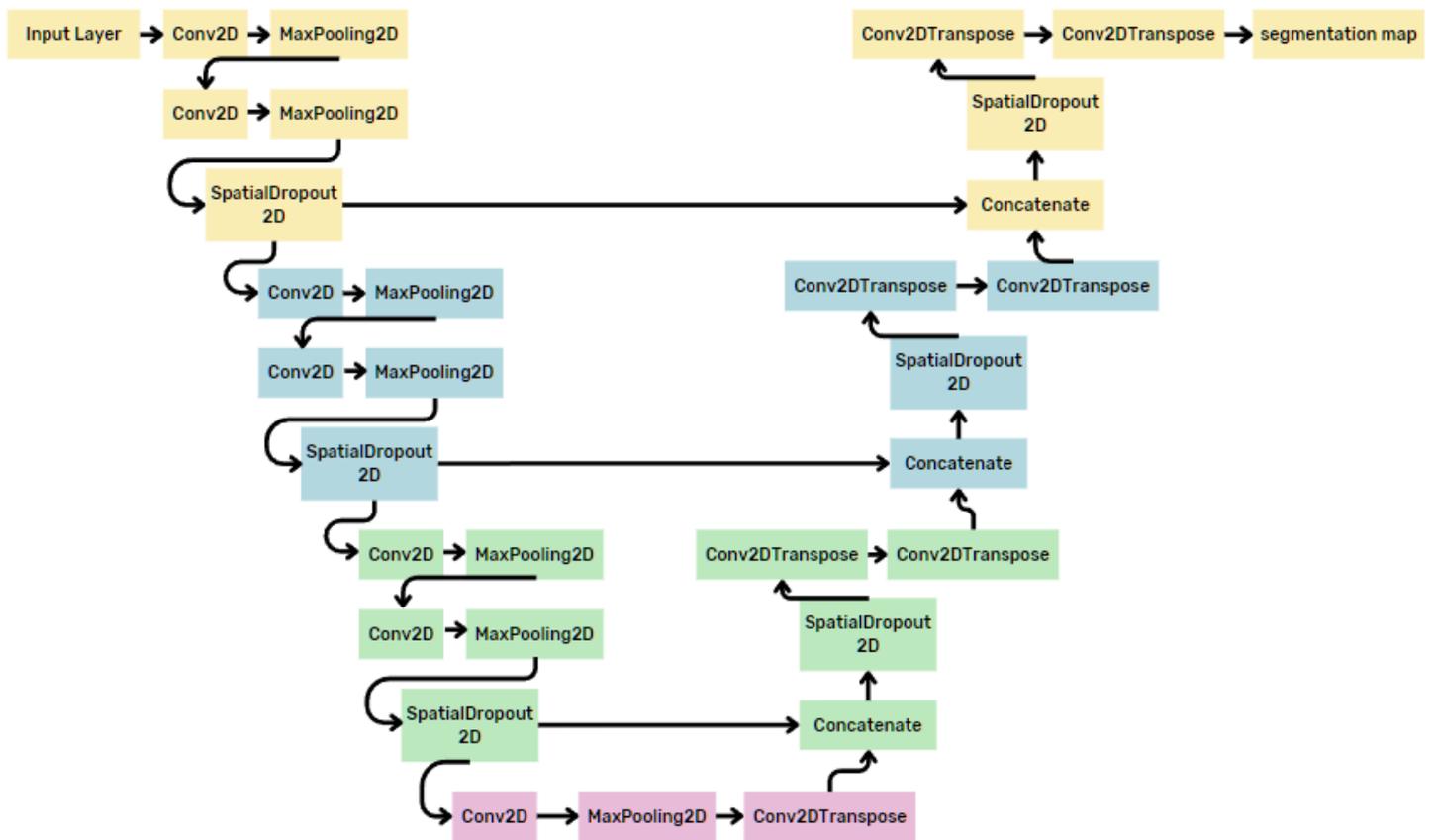


Figura 4.2: Red Neuronal U-Net utilizada para la segmentación

Fuente: Elaboración Propia

Capítulo 5

Resultados

Los hallazgos de este trabajo de investigación se presentan en tres etapas de experimentación: detección y clasificación y conteo.

5.1. Preparación de base de datos

Para este proyecto se han recopilado distintos base de datos de imágenes. Para el caso de la detección de peces se utilizaron 4 base de datos:

1. Deteccion-1 DB: Cuenta con un total de 600 imágenes divididas considerando un 80 % en train y el %20 en prueba dando asi un test (70 imágenes), val (60 imágenes) y train (470 imágenes). Esta base de datos contiene únicamente imágenes de distintas especies de pez fuera del mar. En la detección solo se busca identificar si hay o no un pez en la imagen. Todas las imágenes en esta base de datos se tomaron con la misma iluminación. Un ejemplo de cómo se ven las imágenes del base de datos se aprecia en la [Figura 5.1](#).
2. Deteccion-2 DB: Cuenta con un total de 2400 imágenes divididas considerando un 70 % en train y el %30 en prueba dando asi un test (100 imágenes), val (200 imágenes) y train (2100 imágenes). Esta base de datos es muy variada ya que incluye fotos de peces dentro y fuera del mar, con fondos de color sólido y con modificaciones de color y rotación. En la detección, también se busca identificar si hay o no un pez en la imagen. Las imágenes de este base de datos se ven en la [Figura 5.2](#).
3. VeoxDB v1: Cuenta con un total de 576 imágenes divididas divididas considerando un 60 % en train y el %40 en prueba dando asi un test (130 imágenes), val (115 imágenes) y train (331 imágenes). Esta proporción fue realizada para evaluar el rendimiento de los modelos y realizar ajustes si es necesario. Esta base de datos fue obtenida por la empresa Veox en criaderos de truchas en Puno, Perú. Las imágenes en esta base de datos son de una sola especie de pez y están balanceadas en términos de iluminación cantidad. Las imágenes de este base de datos se ven en la [Figura 5.3](#).
4. VeoxDB v2: Cuenta con un total de 233 imágenes divididas en test (55 imágenes), val (48 imágenes) y train (130 imágenes). En este dataset no se considero una división porcentual debido a que para el entrenamiento solo se consideraron imagenes limpias. Esta base de datos



Figura 5.1: *Ejemplo de Deteccion-1 DB. Peces fuera del mar en superficies solidas*

Fuente: Propia

es una mejora de la anterior, la VeoxDB v1. Contiene imágenes de la misma especie de pez en diferentes posiciones y situaciones. Se eliminaron imágenes de mala calidad y aquellas que contenían mucha distorsión en la identificación de los peces debido a su movimiento. También se balancearon en términos de iluminación y cantidad.

El etiquetado de estos datos para el entrenamiento se representa en cajas que encierran al pez como se ve en la [Figura 5.4](#). Estas cajas son creadas para cada pez y son almacenadas en un archivo '.txt' para cada imagen del base de datos.

Por otro lado para la segmentación de peces se utilizo solo un base de datos:

1. Este base de datos contiene 9 tipos diferentes de peces(dorada, besugo, lubina, salmonete, jurel espadín negro, salmonete rayado, trucha, camarones) obtenidos de un supermercado en Izmir, Turquía. Este base de datos sufrió data augmentation por lo cual para cada tipo de marisco hay 1000 imagenes. El tipo de imagenes se pueden ver en la [Figura 5.5](#) [51].

El etiquetado de este base de datos para el entrenamiento esta hecho por otra imagen como se muestra en la [Figura 5.6](#), de tal forma que se pueda crear una mascara y saber que parte de la imagen es la que se debe segmentar.

Para el caso del conteo se utilizó un modelo ya creado llamado DeepFish, este modelo también es capaz de segmentar y detectar peces. Las imagenes de las que consta este base de datos se pueden ver en la [Figura 5.7](#).



Figura 5.2: *Ejemplo de Deteccion-2 DB. Peces dentro y fuera del agua incluidas laminas de peces*

Fuente: Propia

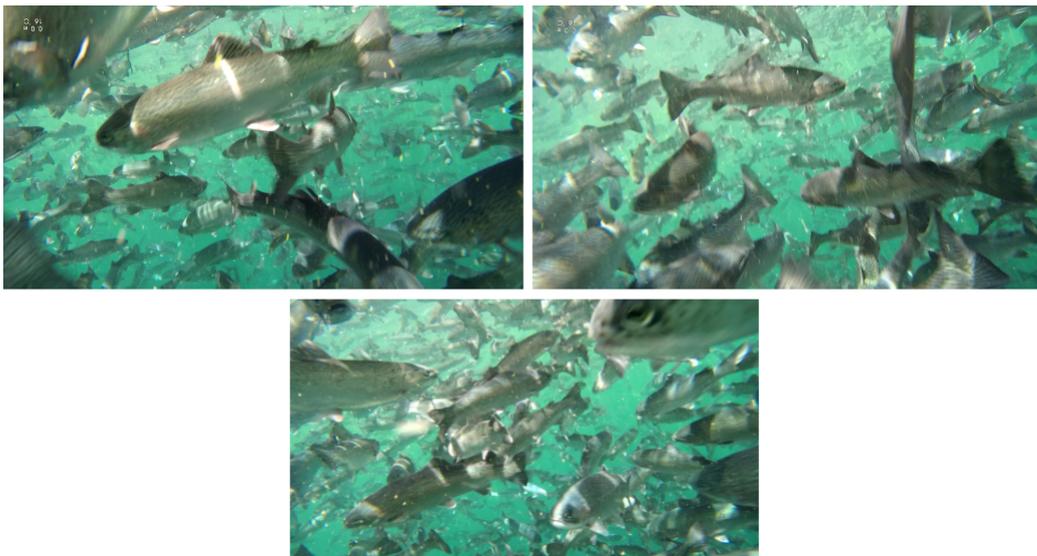


Figura 5.3: *Ejemplo de VeoxDB. Gran cantidad de peces por debajo del agua*

Fuente: Propia

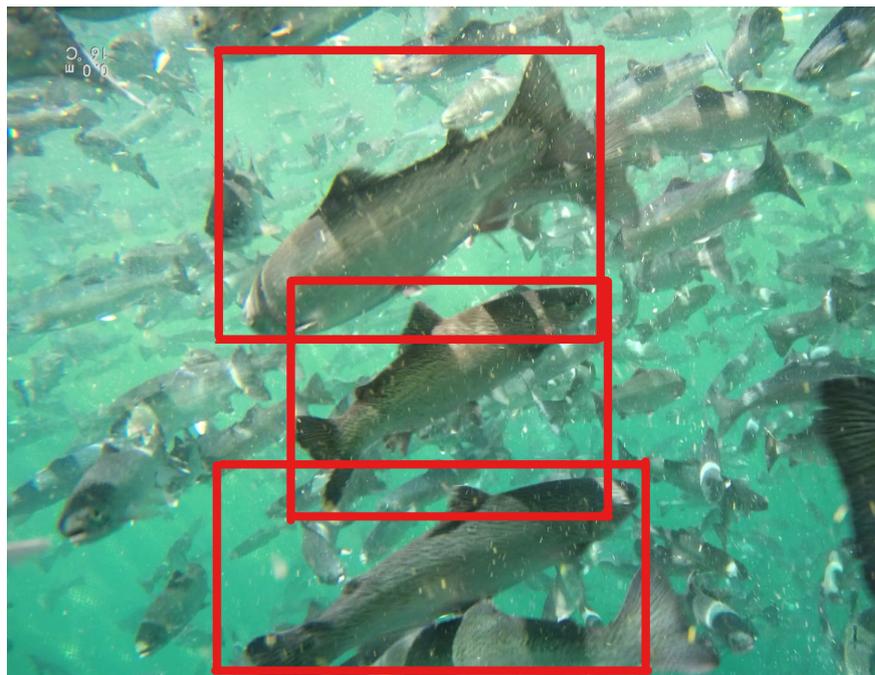


Figura 5.4: Ejemplo de cajas usadas para detección. Enmarcan al pez

Fuente: Propia

5.2. Resultados de desempeño

Respecto a la detección, se procedió a probar los datasets con YOLOv5. Los resultados indican que el modelo tiene mejores resultados en las bases de datos Deteccion-1 y Deteccion-2, con valores de precisión de 0.89 y 0.78, respectivamente, y valores de recall de 0.87 y 0.77, respectivamente. Esto indica que el modelo es bastante preciso y efectivo en la detección de peces en estas bases de datos. Los resultados se observan en la [Tabla 5.1](#).

Cuadro 5.1: Resultados modelo YOLOv5

Detection YOLOv5	Precision	Recall
Deteccion-1 DB	0.89	0.87
Deteccion-2 DB	0.78	0.77
VeoxDB v1	0.59	0.60
VeoxDB v2	0.72	0.66

Fuente: Propia

Respecto a la segmentación, el dataset fue probado con U-Net dando unos resultados prometedores. El valor de 'loss' indica la magnitud del error del modelo durante el entrenamiento. Un valor de 0.05 indica que el modelo ha logrado minimizar el error de segmentación en gran medida. El valor de 'accuracy' indica la precisión del modelo en la tarea de segmentación de peces. Un valor de 97.5% indica que el modelo ha segmentado correctamente el 97.5% de los peces en el conjunto de datos. Los resultados se observan en la [Tabla 5.2](#).

Respecto al conteo los resultados dieron: MAE counting mide el error absoluto medio en la tarea de conteo de peces. Un valor de 0.38 indica que el modelo tiene un error absoluto medio de 0.38 peces en la tarea de conteo.



Figura 5.5: Ejemplo de base de datos segmentación. Imágenes de peces fuera del agua frente a un fondo sólido

Fuente: [51]

Cuadro 5.2: Resultados modelo U-Net

Segmentation - UNET	Loss	Accuracy
Dataset	0.05	97.5 %

Fuente: Propia

MAE localization mide el error absoluto medio en la tarea de localización de peces. Un valor de 0.21 indica que el modelo tiene un error absoluto medio de 0.21 píxeles en la tarea de localización de peces.

GAME localization mide la precisión en la tarea de localización de peces. Un valor de 1.22 indica que el modelo tiene una precisión de 1.22 píxeles en la tarea de localización de peces. Los resultados se observan en la [Tabla 5.3](#).

La evaluación del modelo se realizó mediante la generación de curvas de precisión para cada una de las bases de datos utilizadas. Las curvas de precisión permiten visualizar de manera clara y detallada cómo el modelo se desempeña en términos de precisión a medida que se modifica el umbral de confianza. En la [Figura 5.8](#) esta la curva de precisión obtenida por el Deteccion-1 DB, mientras en la [Figura 5.9](#) esta la curva obtenida por el Deteccion-2 DB. Estas curvas permiten identificar en qué valores de umbral de confianza el modelo alcanza una mayor precisión, lo que es fundamental para establecer los límites adecuados y ajustar el modelo según los requerimientos específicos del usuario.

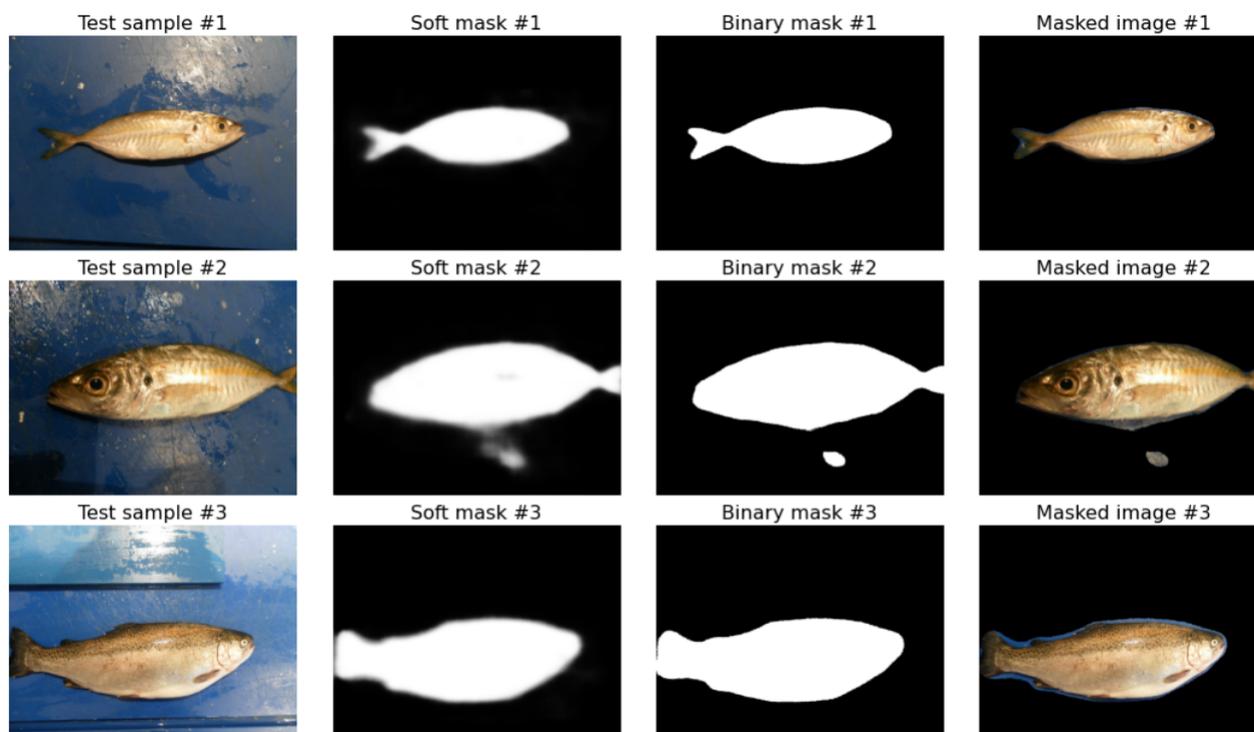


Figura 5.6: Ejemplo de base de datos segmentación. Se encuentra imagen base y mascarará para la segmentación

Fuente: [51]

Cuadro 5.3: Resultados modelo DeepFish

DeepFish	Counting	Localization	
	MAE	MAE	GAME
Dataset DeepFish	0.38	0.21	1.22

Fuente: Propia

Además, se generó una comparativa de las curvas de precisión obtenidas con la base de datos VeoxDB v1 y la VeoxDB v2, en la [Figura 5.10](#) Esta comparativa permite visualizar de manera clara las mejoras que se obtuvieron al limpiar la data de entrada, lo que resulta fundamental para mejorar la precisión del modelo.

También se generó una curva de precisión para la segmentación de objetos. En la [Figura 5.11](#) se presenta la curva de precisión obtenida por la base de datos de segmentación. Esta curva permite evaluar la precisión del modelo para detectar los límites de los objetos y delimitarlos de manera adecuada.

5.3. Discusión

En esta tesis, se presentaron los resultados de la detección, segmentación y conteo de peces utilizando diferentes conjuntos de datos y modelos. A continuación, se discutirán los hallazgos y su relevancia en el contexto de la investigación.

En términos de la detección de peces, se observó que el modelo YOLOv5 obtuvo mejores resultados en las bases de datos Deteccion-1 y Deteccion-2, con valores de precisión de 0.89 y 0.78,



Figura 5.7: Ejemplo de base de datos conteo. Imágenes de peces bajo del mar

Fuente: [50]

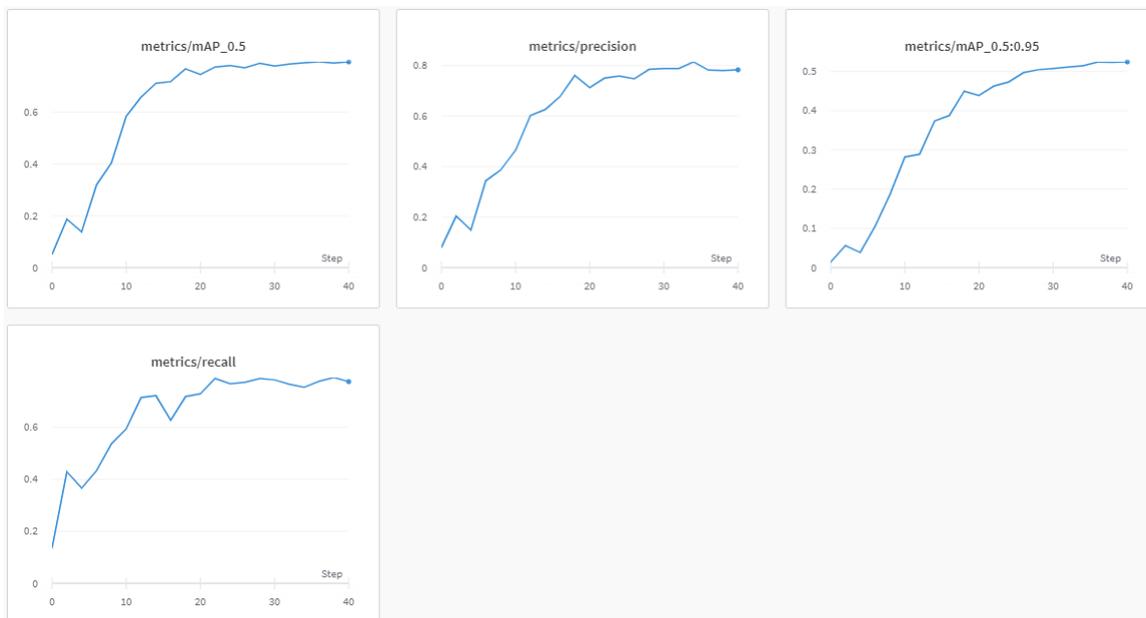


Figura 5.8: Curva de precisión de la base de datos Deteccion-1 DB

Fuente: Propia

respectivamente, y valores de recall de 0.87 y 0.77, respectivamente. Estos resultados indican que el modelo es preciso y efectivo en la detección de peces en estas bases de datos. Sin embargo, se observó una disminución en el desempeño en las bases de datos VeoxDB v1 y VeoxDB v2, con va-

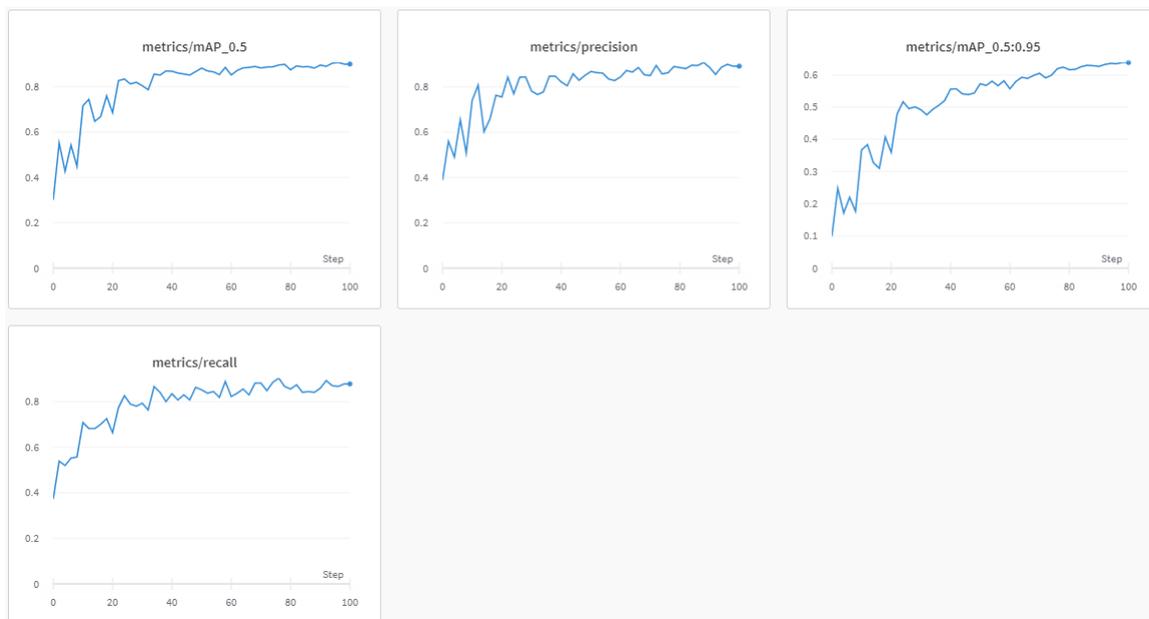


Figura 5.9: Curva de precisión de la base de datos Deteccion-2 DB

Fuente: Propia

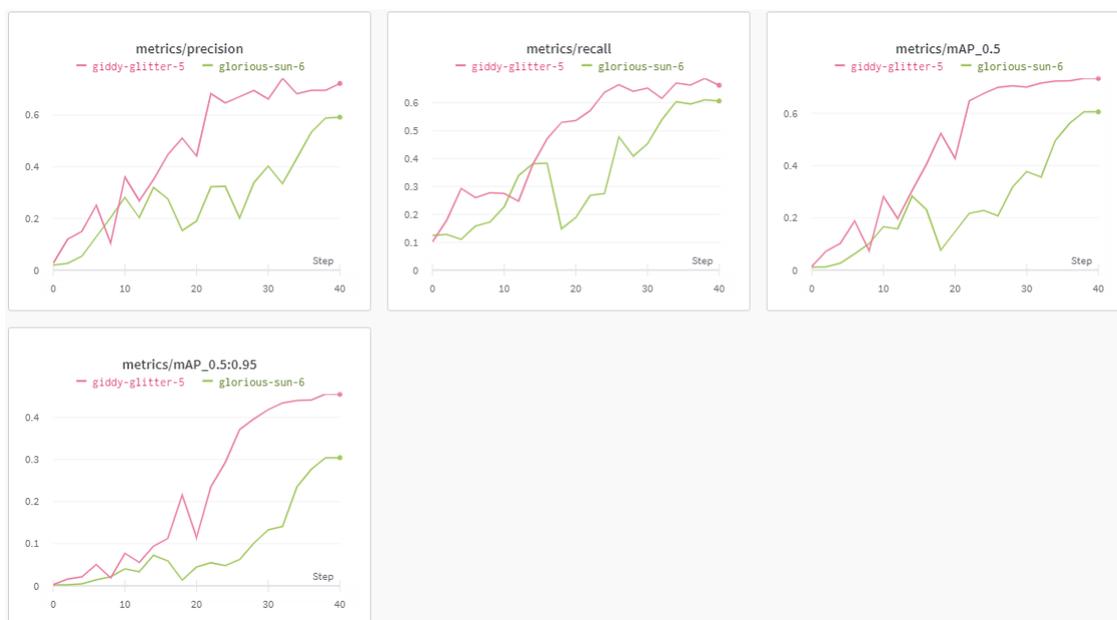


Figura 5.10: Comparación de curva de precisión de la base de datos VeoxDB v1 y v2

Fuente: Propia

lores de precisión de 0.59 y 0.72, respectivamente. Esta diferencia en el desempeño puede atribuirse a la variabilidad en las condiciones de iluminación y la presencia de otros objetos en las imágenes de estas bases de datos.

Es importante destacar que los resultados de la detección de peces son prometedores, ya que la detección precisa de peces es un paso crucial en la identificación y seguimiento de poblaciones acuáticas. Estos resultados pueden tener aplicaciones prácticas en la monitorización de los peces determinando su abundancia y los cambios dentro de la población de peces.

En cuanto a la segmentación de peces, se utilizó el modelo U-Net y se obtuvieron resultados

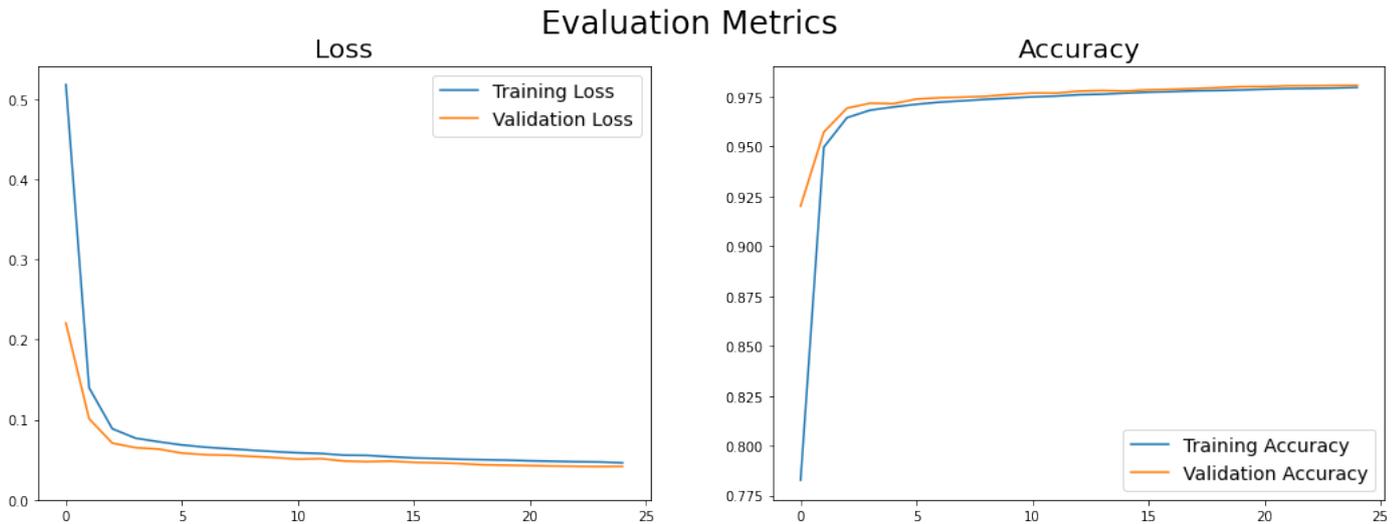


Figura 5.11: Curva de precisión de la base de datos de Segmentación

Fuente: Propia

prometedores en el conjunto de datos utilizado. El valor de 'loss' de 0.05 indica que el modelo ha logrado minimizar el error de segmentación en gran medida. Además, se alcanzó una precisión del 97.5% en la tarea de segmentación de peces. Estos resultados demuestran la capacidad del modelo U-Net para segmentar con precisión los peces en las imágenes y proporcionar información detallada sobre su forma y contorno. Esta información puede ser útil en estudios de morfología y análisis de patrones de coloración en peces.

En relación al conteo de peces, se utilizó el modelo DeepFish, que combina la detección, segmentación y conteo de peces. Los resultados mostraron un error absoluto medio (MAE) de 0.38 en la tarea de conteo de peces y un MAE de 0.21 píxeles en la tarea de localización de peces. Además, se obtuvo un valor de precisión de 1.22 píxeles en la tarea de localización de peces. Estos resultados indican que el modelo DeepFish es capaz de contar y localizar con precisión los peces en las imágenes. Esta capacidad de conteo preciso puede ser valiosa en estudios de población de peces y evaluaciones de stocks pesqueros.

En general, los resultados obtenidos en este estudio demuestran el potencial de los modelos de deep learning en la detección, segmentación y conteo de peces. Estas técnicas pueden proporcionar información detallada sobre la presencia, distribución y características de los peces en diferentes entornos acuáticos. Sin embargo, es importante tener en cuenta las limitaciones y desafíos asociados

Capítulo 6

Conclusiones

En esta investigación, se lograron alcanzar los objetivos propuestos relacionados con la detección, segmentación y conteo de peces utilizando técnicas de aprendizaje profundo. A través de un análisis exhaustivo de diversas técnicas y métodos, se desarrolló una herramienta práctica que permite realizar estas tareas de manera sencilla y eficiente.

Se generó un dataset completo y representativo de imágenes de peces utilizado para entrenar y evaluar los modelos propuestos. Su creación contribuye al desarrollo de técnicas de análisis de peces y servirá como referencia en futuras investigaciones.

En la primera fase de la investigación, se logró implementar con éxito la detección de peces mediante el modelo YOLOv5. Se creó un conjunto de imágenes etiquetadas para el entrenamiento del modelo, lo cual resultó en una precisión del 89

En la etapa de segmentación, se entrenó el modelo U-Net. Este modelo logró obtener una precisión del 97.5

En cuanto al conteo de peces, se utilizó el modelo de DeepFish, el cual mostró una gran exactitud en la tarea. Se logró obtener un Error Absoluto Medio (MAE) del 38

Finalmente, se desarrolló una herramienta práctica y de fácil uso que integra todas las técnicas de detección, segmentación y conteo de peces. Esta herramienta permite realizar estas tareas de forma automatizada y eficiente, lo que facilita el monitoreo y análisis de poblaciones de peces en entornos de crianza.

En conclusión, a través de la investigación y el desarrollo de la herramienta propuesta, se ha logrado avanzar en el campo de la detección, segmentación y conteo de peces utilizando técnicas de aprendizaje profundo. Los resultados obtenidos demuestran la eficacia y precisión de las técnicas implementadas, lo cual contribuye al conocimiento y manejo adecuado de las poblaciones de peces en entornos de crianza.

Bibliografía

- [1] R. Mejía and G. Rosales, “Sistema de detección y clasificación de peces utilizando visión computacional,” in *Innovando la educación en tecnología. Actas del II Congreso Internacional de Ingeniería de Sistemas*. Universidad de Lima, 2020. [Online]. Available: <https://doi.org/10.26439/ciis2019.5507> 1
- [2] “Crianza y producción de truchas — rmr-peru.com,” <https://www.rmr-peru.com/crianza-de-truchas.htm>, [Accessed 18-Oct-2022]. 1
- [3] “Acuicultura: trucha. una opción para el desarrollo de comunidades andinas — hdl.handle.net,” <https://hdl.handle.net/11042/3554>, [Accessed 18-Oct-2022]. 1
- [4] G. Vianna, D. Zeller, and D. Pauly, “Fisheries and policy implications for human nutrition,” *Current Environmental Health Reports*, vol. 7, no. 3, pp. 161–169, 2020. 2
- [5] G. Shobha and S. Rangaswamy, “Machine learning,” in *Handbook of Statistics*. Elsevier, 2018, pp. 197–228. [Online]. Available: <https://doi.org/10.1016/bs.host.2018.07.004> 4, 7
- [6] K. Krishnan, *Data warehousing in the age of big data*, ser. The Morgan Kaufmann Series on Business Intelligence. Oxford, England: Morgan Kaufmann, May 2013. 4
- [7] T. W. Edgar and D. O. Manz, *Research methods for cyber security*. Rockland, MA: Syngress Media, Apr. 2017. 4
- [8] E. Alpaydin, *Introduction to Machine Learning*, 3rd ed., ser. Adaptive Computation and Machine Learning series. London, England: MIT Press, Aug. 2014. 5
- [9] C. Borgelt, “An implementation of the FP-growth algorithm,” in *Proceedings of the 1st international workshop on open source data mining frequent pattern mining implementations - OSDM '05*. ACM Press, 2005. [Online]. Available: <https://doi.org/10.1145/1133905.1133907> 5
- [10] M. A. Mazurowski, M. Buda, A. Saha, and M. R. Bashir, “Deep learning in radiology: An overview of the concepts and a survey of the state of the art with focus on MRI,” *Journal of Magnetic Resonance Imaging*, vol. 49, no. 4, pp. 939–954, Dec. 2018. [Online]. Available: <https://doi.org/10.1002/jmri.26534> 6
- [11] G. Chartrand, P. M. Cheng, E. Vorontsov, M. Drozdal, S. Turcotte, C. J. Pal, S. Kadoury, and A. Tang, “Deep learning: A primer for radiologists,” *RadioGraphics*, vol. 37, no. 7, pp. 2113–2131, Nov. 2017. [Online]. Available: <https://doi.org/10.1148/rg.2017170077> 6, 9

- [12] H. Block, “A review of “perceptrons: An introduction to computational geometry,” *Information and Control*, vol. 17, no. 5, pp. 501–522, Dec. 1970. [Online]. Available: [https://doi.org/10.1016/s0019-9958\(70\)90409-2](https://doi.org/10.1016/s0019-9958(70)90409-2) 6
- [13] C. C. Aggarwal, *Neural Networks and Deep Learning*. Springer International Publishing, 2018. [Online]. Available: <https://doi.org/10.1007/978-3-319-94463-0> 9
- [14] M. Kahng, P. Y. Andrews, A. Kalro, and D. H. Chau, “ActiVis: Visual exploration of industry-scale deep neural network models,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 1, pp. 88–97, Jan. 2018. [Online]. Available: <https://doi.org/10.1109/tvcg.2017.2744718> 9
- [15] Q. Zhang, Y. N. Wu, and S.-C. Zhu, “Interpretable convolutional neural networks,” 2017. [Online]. Available: <https://arxiv.org/abs/1710.00935> 9, 10
- [16] “Cómo funcionan las convolutional neural networks - visión por ordenador,” <https://www.aprendemachinelearning.com/como-funcionan-las-convolutional-neural-networks-vision-por-ordenador/>, 2023. 10
- [17] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal loss for dense object detection,” 2017. [Online]. Available: <https://arxiv.org/abs/1708.02002> 10, 11
- [18] M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes (VOC) challenge,” *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, Sep. 2009. [Online]. Available: <https://doi.org/10.1007/s11263-009-0275-4> 10, 11
- [19] J. Redmon and A. Farhadi, “YOLO9000: Better, faster, stronger,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Jul. 2017. [Online]. Available: <https://doi.org/10.1109/cvpr.2017.690> 11
- [20] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” 2015. [Online]. Available: <https://arxiv.org/abs/1506.01497> 11
- [21] J. Redmon, “Darknet: Open Source Neural Networks in C — pjreddie.com,” <https://pjreddie.com/darknet/>, [Accessed 31-Oct-2022]. 11
- [22] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” 2015. [Online]. Available: <https://arxiv.org/abs/1506.02640> 11
- [23] H.-K. Jung and G.-S. Choi, “Improved YOLOv5: Efficient object detection using drone images under various conditions,” *Applied Sciences*, vol. 12, no. 14, p. 7255, Jul. 2022. [Online]. Available: <https://doi.org/10.3390/app12147255> 11
- [24] Y. Xiao, Z. Tian, J. Yu, Y. Zhang, S. Liu, S. Du, and X. Lan, “A review of object detection based on deep learning,” *Multimedia Tools and Applications*, vol. 79, no. 33-34, pp. 23 729–23 791, Jun. 2020. [Online]. Available: <https://doi.org/10.1007/s11042-020-08976-6> 12
- [25] N. M. Zaitoun and M. J. Aqel, “Survey on image segmentation techniques,” *Procedia Computer Science*, vol. 65, pp. 797–806, 2015. [Online]. Available: <https://doi.org/10.1016/j.procs.2015.09.027> 12, 13

- [26] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” 2015. [Online]. Available: <https://arxiv.org/abs/1505.04597> 13, 23
- [27] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask r-cnn,” 2017. [Online]. Available: <https://arxiv.org/abs/1703.06870> 13
- [28] A. Jalal, A. Salman, A. Mian, M. Shortis, and F. Shafait, “Fish detection and species classification in underwater environments using deep learning with temporal information,” *Ecological Informatics*, vol. 57, p. 101088, May 2020. [Online]. Available: <https://doi.org/10.1016/j.ecoinf.2020.101088> 15
- [29] H. Wu, S. He, Z. Deng, L. Kou, K. Huang, F. Suo, and Z. Cao, “Fishery monitoring system with AUV based on YOLO and SGBM,” in *2019 Chinese Control Conference (CCC)*. IEEE, Jul. 2019. [Online]. Available: <https://doi.org/10.23919/chicc.2019.8866087> 15
- [30] H. E.-D. Mohamed, A. Fadel, O. Anas, Y. Wageeh, N. ElMasry, A. Nabil, and A. Atia, “MSR-YOLO: Method to enhance fish detection and tracking in fish farms,” *Procedia Computer Science*, vol. 170, pp. 539–546, 2020. [Online]. Available: <https://doi.org/10.1016/j.procs.2020.03.123> 15
- [31] J.-H. Park and C. Kang, “A study on enhancement of fish recognition using cumulative mean of YOLO network in underwater video images,” *Journal of Marine Science and Engineering*, vol. 8, no. 11, p. 952, Nov. 2020. [Online]. Available: <https://doi.org/10.3390/jmse8110952> 15
- [32] K. Raza and S. Hong, “Fast and accurate fish detection design with improved YOLO-v3 model and transfer learning,” *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 2, 2020. [Online]. Available: <https://doi.org/10.14569/ijacsa.2020.0110202> 15
- [33] B. Zion, A. Shklyar, and I. Karplus, “Sorting fish by computer vision,” *Computers and Electronics in Agriculture*, vol. 23, no. 3, pp. 175–187, Sep. 1999. [Online]. Available: [https://doi.org/10.1016/S0168-1699\(99\)00030-7](https://doi.org/10.1016/S0168-1699(99)00030-7) 16
- [34] E. Prasetyo, N. Suciati, and C. Fatichah, “Yolov4-tiny and spatial pyramid pooling for detecting head and tail of fish,” in *2021 International Conference on Artificial Intelligence and Computer Science Technology (ICAICST)*. IEEE, Jun. 2021. [Online]. Available: <https://doi.org/10.1109/icaicst53116.2021.9497822> 16
- [35] J. Betancourt, W. Coral, and J. Colorado, “An integrated ROV solution for underwater net-cage inspection in fish farms using computer vision,” *SN Applied Sciences*, vol. 2, no. 12, Nov. 2020. [Online]. Available: <https://doi.org/10.1007/s42452-020-03623-z> 16
- [36] B. S. Rekha, G. N. Srinivasan, S. K. Reddy, D. Kakwani, and N. Bhattad, “Fish detection and classification using convolutional neural networks,” in *Computational Vision and Bio-Inspired Computing*. Springer International Publishing, 2020, pp. 1221–1231. [Online]. Available: https://doi.org/10.1007/978-3-030-37218-7_128 16
- [37] S. Cui, Y. Zhou, Y. Wang, and L. Zhai, “Fish detection using deep learning,” *Applied Computational Intelligence and Soft Computing*, vol. 2020, pp. 1–13, Jan. 2020. [Online]. Available: <https://doi.org/10.1155/2020/3738108> 16

- [38] G. F. Garcia, F. Martignac, M. Nevoux, L. Beaulaton, and T. Corpetti, “A deep neural network for multi-species fish detection using multiple acoustic cameras,” 2021. [Online]. Available: <https://arxiv.org/abs/2109.10664> 16
- [39] E. Prasetyo, N. Suciati, and C. Fatichah, “A comparison of YOLO and mask r-CNN for segmenting head and tail of fish,” in *2020 4th International Conference on Informatics and Computational Sciences (ICICoS)*. IEEE, Nov. 2020. [Online]. Available: <https://doi.org/10.1109/icicos51170.2020.9299024> 16
- [40] A. Salman, S. A. Siddiqui, F. Shafait, A. Mian, M. R. Shortis, K. Khurshid, A. Ulges, and U. Schwanecke, “Automatic fish detection in underwater videos by a deep neural network-based hybrid motion learning system,” *ICES Journal of Marine Science*, vol. 77, no. 4, pp. 1295–1307, Feb. 2019. [Online]. Available: <https://doi.org/10.1093/icesjms/fsz025> 17
- [41] J. C. Ovalle, C. Vilas, and L. T. Antelo, “On the use of deep learning for fish species recognition and quantification on board fishing vessels,” *Marine Policy*, vol. 139, p. 105015, May 2022. [Online]. Available: <https://doi.org/10.1016/j.marpol.2022.105015> 17
- [42] R. Garcia, R. Prados, J. Quintana, A. Tempelaar, N. Gracias, S. Rosen, H. Vågstøl, and K. Løvall, “Automatic segmentation of fish using deep learning with application to fish size measurement,” *ICES Journal of Marine Science*, vol. 77, no. 4, pp. 1354–1366, Oct. 2019. [Online]. Available: <https://doi.org/10.1093/icesjms/fsz186> 17
- [43] N. E. Garcia-D'Urso, A. Galan-Cuenca, P. Climent-Perez, M. Saval-Calvo, J. Azorin-Lopez, and A. Fuster-Guillo, “Efficient instance segmentation using deep learning for species identification in fish markets,” in *2022 International Joint Conference on Neural Networks (IJCNN)*. IEEE, Jul. 2022. [Online]. Available: <https://doi.org/10.1109/ijcnn55064.2022.9892945> 17
- [44] N. F. F. Alshdaifat, A. Z. Talib, and M. A. Osman, “Improved deep learning framework for fish segmentation in underwater videos,” *Ecological Informatics*, vol. 59, p. 101121, Sep. 2020. [Online]. Available: <https://doi.org/10.1016/j.ecoinf.2020.101121> 17
- [45] S. Zhang, X. Yang, Y. Wang, Z. Zhao, J. Liu, Y. Liu, C. Sun, and C. Zhou, “Automatic fish population counting by machine vision and a hybrid deep neural network model,” *Animals*, vol. 10, no. 2, p. 364, Feb. 2020. [Online]. Available: <https://doi.org/10.3390/ani10020364> 17
- [46] C.-H. Tseng and Y.-F. Kuo, “Detecting and counting harvested fish and measuring fish body lengths in video using deep learning methods,” in *2019 Boston, Massachusetts July 7- July 10, 2019*. American Society of Agricultural and Biological Engineers, 2019. [Online]. Available: <https://doi.org/10.13031/aim.201900408> 17
- [47] V. Kandimalla, M. Richard, F. Smith, J. Quirion, L. Torgo, and C. Whidden, “Automated detection, classification and counting of fish in fish passages with deep learning,” *Frontiers in Marine Science*, vol. 8, Jan. 2022. [Online]. Available: <https://doi.org/10.3389/fmars.2021.823173> 17
- [48] S. Schneider and A. Zhuang, “Counting fish and dolphins in sonar images using deep learning,” 2020. [Online]. Available: <https://arxiv.org/abs/2007.12808> 18

- [49] Y. Wageeh, H. E.-D. Mohamed, A. Fadl, O. Anas, N. ElMasry, A. Nabil, and A. Atia, “YOLO fish detection with euclidean tracking in fish farms,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 1, pp. 5–12, Jan. 2021. [Online]. Available: <https://doi.org/10.1007/s12652-020-02847-6> 18
- [50] A. Saleh, I. H. Laradji, D. A. Konovalov, M. Bradley, D. Vazquez, and M. Sheaves, “A realistic fish-habitat dataset to evaluate algorithms for underwater visual analysis,” 2020. 23, 31
- [51] O. Ulucan, D. Karakaya, and M. Turkan, “A large-scale dataset for fish segmentation and classification,” in *2020 Innovations in Intelligent Systems and Applications Conference (ASYU)*. IEEE, 2020, pp. 1–5. 26, 29, 30