



Universidad  
Nacional  
de Loja

# Universidad Nacional de Loja

## Facultad de la Energía, las Industrias y los Recursos Naturales No Renovables

### Carrera de Computación

**Desarrollo de un Dataset Especializado para la Detección Automática de Sacos de Balanceado de Camarón Aquaxcel GROWER 5 (25 kg), Aquaxcel Pre-Starter 2 (25 kg) y ECOFEED Purina (25 kg) utilizando el Modelo YOLOv8n**

**Development of a Specialized Dataset for the Automatic Detection of Aquaxcel GROWER 5 (25 kg), Aquaxcel Pre-Starter 2 (25 kg) and ECOFEED Purina (25 kg) Shrimp Feed Bags using the YOLOv8n Model**

Trabajo de Integración Curricular, previo a la obtención del título de Ingeniero en Ciencias de la Computación.

**AUTOR:**

Keyner Alexis Manchay Montoya

**DIRECTOR:**

Ing. Oscar Miguel Cumbicus Pineda Mg.Sc

Loja – Ecuador

2024

## Certificación

Loja, 09 de Septiembre de 2024

Ing. Oscar Cumbicus Mg. Sc.

**DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR**

### **CERTIFICO:**

Que he revisado y orientado todo el proceso de la elaboración del Trabajo de Integración Curricular denominado: **Desarrollo de un Dataset Especializado para la Detección Automática de Sacos de Balanceado de Camarón Aquaxcel GROWER 5 (25 kg), Aquaxcel Pre-Starter 2 (25 kg) y ECOFEED Purina (25 kg) utilizando el Modelo YOLOv8n**, previo a la obtención del título de **Ingeniero en Ciencias de la Computación** de autoría del estudiante: **Keyner Alexis Manchay Montoya** con **cédula de identidad Nro. 0706844255**, una vez que el trabajo cumple con todos los requisitos exigidos por la Universidad Nacional de Loja, para el efecto, autorizo la presentación del mismo para su respectiva sustentación y defensa.

Ing. Oscar Cumbicus Mg. Sc.

**DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR**

## **Autoría**

Yo, **Keyner Alexis Manchay Montoya**, declaro ser autor del presente Trabajo de Integración Curricular y eximo expresamente a la Universidad Nacional de Loja y a sus representantes jurídicos, de posibles reclamos y acciones legales, por el contenido del mismo. Adicionalmente, acepto y autorizo a la Universidad Nacional de Loja la publicación de mi Trabajo de Integración Curricular, en el Repositorio Digital Institucional – Biblioteca Virtual.

**Firma:**

**Cédula de identidad:** Keyner Alexis Manchay Montoya

**Fecha:** 09/09/2024

**Correo electrónico:** keyner.manchay@unl.edu.ec

**Teléfono:** +593 991 869 497

**Carta de autorización por parte del autor, para consulta, reproducción parcial o total y/o publicación electrónica del texto completo, del Trabajo de Integración Curricular**

Yo, **Keyner Alexis Manchay Montoya**, declaro ser el autor del Trabajo de Integración Curricular denominado: **Desarrollo de un Dataset Especializado para la Detección Automática de Sacos de Balanceado de Camarón Aquaxcel GROWER 5 (25 kg), Aquaxcel Pre-Starter 2 (25 kg) y ECOFEED Purina (25 kg) utilizando el Modelo YOLOv8n**, autorizo al sistema Bibliotecario de la Universidad Nacional de Loja para que, con fines académicos, muestre la producción intelectual de la Universidad, a través de la visibilidad de su contenido en el Repositorio Institucional.

Los usuarios pueden consultar el contenido de este trabajo en el Repositorio Institucional, en las redes de información del país y del exterior con las cuales tenga convenio la Universidad.

La Universidad Nacional de Loja, no se responsabiliza por el plagio o copia del Trabajo de Integración Curricular que realice un tercero.

Para constancia de esta autorización, suscribo, en la ciudad de Loja, a los nueve días del mes de septiembre del dos mil veinticuatro.

**Firma:**

**Autor:** Keyner Alexis Manchay Montoya

**Cédula de identidad:** 0706844255

**Dirección:** Loja, Loja Ecuador

**Correo electrónico:** keyner.manchay@unl.edu.ec

**Teléfono:** +593 991 869 497

**DATOS COMPLEMENTARIOS:**

**Director del Trabajo de Integración Curricular:** Ing. Oscar Miguel Cumbicus Pineda Mg. Sc.

## **Dedicatoria**

Esta investigación se lo dedico a mis seres queridos, a mis padres, abuelos, hermanos y familia, con sus nombres Anita Montoya, Rene Manchay, Anita Cruz, Manuel Montoya, Marcelo Manchay y Jazmin Montoya, quienes fueron los que me apoyaron, permitiendo poder superarme en mis objetivos académicos y me impulsaron a continuar.

***Keyner Alexis Manchay Montoya***

## **Agradecimiento**

Le agradezco a las personas que me brindaron su ayuda para poder completar este trabajo de integración curricular (TIC), a mi director Ing. Oscar Miguel Cumbicus Pineda Mg. Sc., por la guía y conocimiento brindado en esta investigación, a la administradora de contabilidad Ing. Marjorie Aponte Suárez de la empresa de Asociación de Productores Camaroneros (ASOCAM) por facilitar la información requerida a lo largo de la investigación, también a los profesores de la universidad nacional de Loja (UNL), por sus conocimientos compartidos, y a mi familia por su apoyo durante todo este proceso.

***Keyner Alexis Manchay Montoya***

## Índice de Contenidos

<b>Portada</b> .....	<b>i</b>
<b>Certificación</b> .....	<b>ii</b>
<b>Autoría</b> .....	<b>iii</b>
<b>Carta de autorización</b> .....	<b>iv</b>
<b>Dedicatoria</b> .....	<b>v</b>
<b>Agradecimiento</b> .....	<b>vi</b>
<b>Índice de Contenidos</b> .....	<b>vii</b>
Índice de Tablas.....	x
Índice de Figuras.....	xi
Índice de Anexos.....	xiii
<b>1. Título</b> .....	<b>1</b>
<b>2. Resumen</b> .....	<b>2</b>
Abstract.....	3
<b>3. Introducción</b> .....	<b>4</b>
<b>4. Marco teórico</b> .....	<b>6</b>
4.1. Antecedentes.....	6
4.2. Fundamentación Teórica.....	6
4.2.1. Inteligencia Artificial.....	6
4.2.2. Machine Learning.....	7
4.2.3. Deep Learning.....	7
4.2.4. Sistemas Inteligentes.....	7
4.2.5. Visión por Computadora.....	8
4.2.6. Fine Tuning.....	8
4.2.7. Anotación.....	8
4.2.8. Detección de Objetos.....	9
4.2.9. Conjunto de Datos (Dataset).....	10
4.2.10. Sacos de Balanceado de camarón.....	11
4.3. Metodologías y Prácticas en Ciencia de Datos y Machine Learning.....	12
4.3.1. Metodología CRISP-DM.....	12

4.3.2. Prácticas de MLOps.....	13
4.4. Tecnologías.....	14
4.4.1. Google Drive.....	14
4.4.2. Google Colaboratory.....	15
4.4.3. Labellmg.....	15
4.4.4. Visual Studio Code.....	15
4.4.5. Github.....	15
4.4.6. Python.....	16
4.4.7. OpenCV.....	16
4.5. Modelo de Red Convolutacional para Detección de Objetos.....	16
4.5.1. YOLOv8n.....	16
4.6. Trabajos Relacionados.....	17
<b>5. Metodología.....</b>	<b>20</b>
5.1. Área de estudio.....	20
5.2. Procedimiento.....	21
5.2.1. Objetivo 1: Recolectar imágenes de sacos de balanceado de camarón Aquaxcel GROWER 5 (25 kg), Aquaxcel Pre-Starter 2 (25 kg) y ECOFEED Purina (25 kg) para anotarlas y formar un dataset diverso y representativo, tomando como referencia la metodología CRISP-DM y las prácticas de MLOps.....	21
5.2.2. Objetivo 2: Utilizar el modelo YOLOv8n con el dataset creado para medir la precisión y la tasa de error en la detección de sacos de balanceado de camarón Aquaxcel GROWER 5 (25 kg), Aquaxcel Pre-Starter 2 (25 kg), y ECOFEED Purina (25 kg).....	28
5.3. Recursos.....	29
5.3.1. Recursos Metodológicos.....	29
5.3.1.1. Metodología Experimental.....	29
5.3.1.2. Ajuste Fino (Fine Tuning).....	30
5.3.2. Recursos técnicos.....	30
5.3.2.1. Entrevista a la empleada de la empresa ASOCAM.....	30
5.3.2.2. Entrevista al experto en Datasets.....	30
5.3.3. Recursos Tecnológicos.....	31

5.3.3. Recursos Tecnológicos.....	31
5.3.3.1. Hardware.....	31
5.3.3.2. Software.....	31
5.4. Participantes.....	32
<b>6. Resultados.....</b>	<b>33</b>
6.1. Objetivo 1: Recolectar imágenes de sacos de balanceado de camarón Aquaxcel GROWER 5 (25 kg), Aquaxcel Pre-Starter 2 (25 kg) y ECOFEED Purina (25 kg) para anotarlas y formar un dataset diverso y representativo, tomando como referencia la metodología CRISP-DM y las prácticas de MLOps.....	33
6.1.1. Tarea 1: Captura de datos mediante fotografías a los sacos de balanceado de camarón en diversas condiciones y entornos.....	33
6.1.2. Tarea 2: Preparación de los datos mediante la clasificación, criterios de inclusión y exclusión, limpieza y redimensionamiento.....	34
6.1.3. Tarea 3: Anotación de los datos con la herramienta LabelImg.....	41
6.1.4. Tarea 4: División del conjunto de datos entre entrenamiento (train), validación (val) y prueba (test).....	43
6.1.5. Tarea 5: Entrenamiento del modelo YOLOv8n.....	45
6.1.6. Tarea 6: Mejora del conjunto de datos.....	49
6.2. Objetivo 2: Utilizar el modelo YOLOv8n con el dataset creado para medir la precisión y la tasa de error en la detección de sacos de balanceado de camarón Aquaxcel GROWER 5 (25 kg), Aquaxcel Pre-Starter 2 (25 kg), y ECOFEED Purina (25 kg).....	53
<b>7. Discusión.....</b>	<b>61</b>
7.1. Primer objetivo: Recolectar imágenes de sacos de balanceado de camarón Aquaxcel GROWER 5 (25 kg), Aquaxcel Pre-Starter 2 (25 kg) y ECOFEED Purina (25 kg) para anotarlas y formar un dataset diverso y representativo, tomando como referencia la metodología CRISP-DM y las prácticas de MLOps.....	61
7.2. Segundo objetivo: Utilizar el modelo YOLOv8n con el dataset creado para medir la precisión y la tasa de error en la detección de sacos de balanceado de camarón Aquaxcel GROWER 5 (25 kg), Aquaxcel Pre-Starter 2 (25 kg), y ECOFEED Purina (25 kg).....	62
<b>8. Conclusiones.....</b>	<b>64</b>
<b>9. Recomendaciones.....</b>	<b>65</b>
<b>10. Bibliografía.....</b>	<b>67</b>
<b>11. Anexos.....</b>	<b>72</b>

## Índice de Tablas:

<b>Tabla 1.</b> Trabajos relacionados con el tema de investigación del presente trabajo TIC.....	17
<b>Tabla 2.</b> Recursos utilizados para la recolección de datos.....	22
<b>Tabla 3.</b> Condiciones tomadas para la realización de la captura de datos.....	23
<b>Tabla 4.</b> Criterios de inclusión y exclusión para los datos recolectados.....	24
<b>Tabla 5.</b> Requisitos de los modelos de YOLOv8, probados con el dataset COCO.....	25
<b>Tabla 6.</b> Cantidad de datos totales según su clase, después de la eliminación de datos.....	35
<b>Tabla 7.</b> Cantidad total del conjunto de datos, después de la limpieza.....	37
<b>Tabla 8.</b> Índices generados al crear las clases dentro de la herramienta LabelImg.....	42
<b>Tabla 9.</b> División de los datos según su subconjunto.....	45
<b>Tabla 10.</b> Configuración de los hiperparámetros para el entrenamiento de los datos con el modelo yolov8n.pt.....	46
<b>Tabla 11.</b> Rendimiento del Modelo YOLOv8n durante el primer entrenamiento.....	47
<b>Tabla 12.</b> Datos actualizados del conjunto de datos después de la eliminación de imágenes de la mejora del conjunto de datos.....	50
<b>Tabla 13.</b> Rendimiento del Modelo YOLOv8n durante el segundo entrenamiento con el conjunto de datos mejorado.....	51
<b>Tabla 14.</b> Métricas de rendimiento del modelo YOLOv8n probado con el subconjunto de datos de prueba (test).....	53
<b>Tabla 15.</b> Valores de las métricas asociadas a la matriz de confusión del modelo YOLOv8n entrenado.....	60

## Índice de Figuras:

<b>Figura 1.</b> Aplicación de la detección de objeto para una cámara vigilancia para que capte carros (Car).....	10
<b>Figura 2.</b> Ejemplos de diferentes conjuntos de imágenes populares actualmente, realizada por el autor del presente TIC.....	11
<b>Figura 3.</b> Diagrama de flujo de la metodología CRISP-DM.....	13
<b>Figura 4.</b> Elementos que consideran las prácticas de MLOps.....	14
<b>Figura 5.</b> Arquitectura de Red del framework YOLOV8.....	17
<b>Figura 6.</b> Mapa de la ubicación de ASOCAM en la parroquia Hualtaco, provincia del Oro, Ecuador.....	21
<b>Figura 7.</b> Parte de las instalaciones de la empresa ASOCAM. (a) Vista en dirección a la oficina de Ventas. (b) Vista en dirección al bar y productos acuícolas. (c) Vista en dirección a la entrada a la empresa y salida de despacho vía terrestre. (d) Vista en dirección a la salida de despacho vía marítima.....	22
<b>Figura 8.</b> Conjunto de datos de imágenes recolectados de Sacos de balanceado de camarón.....	34
<b>Figura 9.</b> Imagen descartada por no pertenecer a ninguna clase.....	35
<b>Figura 10.</b> Imágenes que no cumplen con los criterios de inclusión y exclusión. (a) Imagen muy borrosa que no se distinguen las características del saco. (b) Imagen borrosa y donde el forro de plástico no ayuda a poder visualizar las características del saco. (c) Imagen donde el cuerpo del saco objetivo no puede verse al menos un 75%.....	36
<b>Figura 11.</b> Gráfico que muestra la cantidad de las Imágenes, según su clase.....	37
<b>Figura 12.</b> Porción de código que establece las dimensiones de la imagen según el modelo YOLOv8n.....	38
<b>Figura 13.</b> Porción de código que redimensiona las imágenes del conjunto de datos.....	39
<b>Figura 14.</b> Presentación en un visualizador de imágenes del resultado del redimensionamiento del conjunto de datos.....	39
<b>Figura 15.</b> Presentación de la declaración de las variables como carpetas de las clases y el contador.....	40
<b>Figura 16.</b> Presentación del script utilizado para el renombramiento de los datos.....	40
<b>Figura 17.</b> Presentación desde la carpeta del conjunto de datos renombrados.....	41
<b>Figura 18.</b> Archivo de anotación “Imagen_566” en formato YOLO (txt).....	42
<b>Figura 19.</b> Imágenes con objetos dentro de cuadros delimitadores. (a) Se etiquetaron dos objetos de la clase 1. (b) Se etiquetaron tres objetos de la clase 2. (c) Se etiquetaron cinco	

objetos de la clase 3.....	43
<b>Figura 20.</b> División del conjunto de datos.....	44
<b>Figura 21.</b> Script para la división de datos en 3 subconjuntos: train (80%), val (10%) y test (10%).....	44
<b>Figura 22.</b> Diagrama del modelo YOLOv8 para la detección de sacos de balanceado de camarón.....	45
<b>Figura 23.</b> Archivo yaml, donde se declaran las rutas de los subconjuntos de datos, número de clases y los nombres de las clases.....	47
<b>Figura 24.</b> Cantidad de instancias en cada clase y la acumulación de las cajas delimitadoras del conjunto de datos después del primer entrenamiento del modelo.....	48
<b>Figura 25.</b> Matriz de confusión normalizada para la clasificación de sacos de balanceado de camarón utilizando el modelo YOLOv8n del primer entrenamiento.....	49
<b>Figura 26.</b> Modificación del código de renombramiento de la tarea 2, para el renombramiento de imágenes junto con sus etiquetas.....	50
<b>Figura 27.</b> División del conjunto de datos mejorado.....	51
<b>Figura 28.</b> Cantidad de instancias en cada clase y la acumulación de las cajas delimitadoras del conjunto de datos mejorado después del segundo entrenamiento del modelo.....	52
<b>Figura 29.</b> Matriz de confusión normalizada para la clasificación de sacos de balanceado de camarón utilizando el modelo YOLOv8n del segundo entrenamiento y el conjunto de datos mejorado.....	52
<b>Figura 30.</b> Porción de código que obtiene respuestas sobre el rendimiento del dataset creado.....	53
<b>Figura 31.</b> Matriz de confusión normalizada del rendimiento del mejor modelo para la clasificación de sacos de balanceado de camarón utilizando el modelo YOLOv8n.....	55
<b>Figura 32.</b> Gráfica de Precision-Confidence Curve para la Clasificación de Sacos de Balanceado de Camarón utilizando el Modelo YOLOv8n.....	56
<b>Figura 33.</b> Gráfica de Precision-Recall Curve para la Clasificación de Sacos de Balanceado de Camarón utilizando el Modelo YOLOv8n.....	57
<b>Figura 34.</b> Gráfica de Recall-Confidence Curve para la Clasificación de Sacos de Balanceado de Camarón utilizando el Modelo YOLOv8n.....	58
<b>Figura 35.</b> Matriz de confusión no normalizado del rendimiento del mejor modelo para la clasificación de sacos de balanceado de camarón utilizando el modelo YOLOv8n.....	58

## Índice de Anexos:

<b>Anexo 1.</b> Entrevista sobre el desarrollo de conjuntos de datos para la detección de objetos... .....	72
<b>Anexo 2.</b> Entrevista aplicada a empleada de la empresa ASOCAM.....	75
<b>Anexo 3.</b> Permiso de autorización para captura de fotografía dentro de las instalaciones de la empresa ASOCAM.....	78
<b>Anexo 4.</b> Certificado de la traducción del resumen del proyecto TIC.....	79

## **1. Título**

**Desarrollo de un Dataset Especializado para la Detección Automática de Sacos de Balanceado de Camarón Aquaxcel GROWER 5 (25 kg), Aquaxcel Pre-Starter 2 (25 kg) y ECOFEED Purina (25 kg) utilizando el Modelo YOLOv8n.**

**Development of a Specialized Dataset for the Automatic Detection of Aquaxcel GROWER 5 (25 kg), Aquaxcel Pre-Starter 2 (25 kg) and ECOFEED Purina (25 kg) Shrimp Feed Bags using the YOLOv8n Model**

## 2. Resumen

Los procesos automatizados en el área logística son necesarios para mantener la eficiencia y competitividad a nivel operativo dentro de una empresa, en la industria acuícola es uno de los sectores que es necesario esta implementación. El presente trabajo tiene el objetivo de crear un conjunto de datos especializado en productos acuícolas, para su utilización con el modelo YOLOv8n para la detección automática de alimentos acuícolas empaquetados en sacos de polipropileno tejido; la recolección de los datos se realizaron en la parroquia Hualtaco, El Oro, Ecuador, en la empresa ASOCAM, consiguiendo imágenes de tres tipos de sacos, tomando condiciones de captura como la iluminación y ángulos diversos, asegurando su calidad con criterios de inclusión y exclusión, para posteriormente ser anotado con cuadros delimitadores y dividido en subconjuntos de entrenamiento, validación y prueba. El modelo YOLOv8n se entrenó en Google Colab, implementando mejoras para equilibrar las clases de datos, resultando en 2460 imágenes para entrenamiento, 307 para validación y 308 para prueba. Los hiperparámetros utilizados para el entrenamiento fueron task (detect), mode (train), data (/content/data1/data.yaml), model (yolov8n.pt), imgsz (640), batch (16). El conjunto de datos es entrenado con el modelo YOLOv8n, demostrando una precisión promedio (mAP) del 81.08% en el umbral (IoU) entre 0.50 y 0.95, además de un 89.02% para un IoU de 0.50, y una precisión global de 89.02%, que se pueden comparar con estudios similares al proyecto, como el empleo en entornos industriales. La tasa de error global que resultó de la evaluación proporciona un 0.0813, demostrando un rendimiento aceptable para mejorar los procesos logísticos con modelos de IA, además de facilitar una base para futuras investigaciones y aplicaciones dentro de la industria. Este proyecto muestra la construcción de un conjunto de datos de imágenes para modernizar procesos industriales, con el modelo YOLOv8n para la detección de objetos con rapidez y precisión.

**Palabras Clave:** Datos de imágenes, Visión por computador, Detección de objetos, Red neuronal convolucional, YOLO.

## **Abstract**

Automated processes in the logistics area are necessary to maintain efficiency and competitiveness at the operational level within a company, and the aquaculture industry is one of the sectors where this implementation is necessary. The objective of this work is to create a data set specialized in aquaculture products to be used with the YOLOv8n model for the automatic detection of aquaculture food packaged in woven polypropylene bags. Data collection was carried out in the parish of Hualtaco, El Oro, Ecuador, by the company ASOCAM, obtaining images of three types of bags, taking capture conditions such as lighting and different angles, ensuring their quality with inclusion and exclusion criteria, to be subsequently annotated with bounding boxes and divided into subsets for training, validation, and testing. The YOLOv8n model was trained in Google Colab, implementing improvements to balance the data classes, resulting in 2460 images for training, 307 for validation, and 308 for testing. The hyperparameters used for training were task (detect), mode (train), data (/content/data1/data.yaml), model (yolov8n.pt), imgsz (640), and batch (16). The dataset is trained with the YOLOv8n model, demonstrating an average accuracy (mAP) of 81.08% at the threshold (IoU) between 0.50 and 0.95, in addition to 89.02% for an IoU of 0.50 and an overall accuracy of 89.02%, which can be compared with studies similar to the project, such as employment in industrial environments. The overall error rate resulting from the evaluation is 0.0813, demonstrating acceptable performance for improving logistics processes with AI models as well as providing a basis for future research and applications within the industry. This project shows the construction of an image dataset for modernizing industrial processes with the YOLOv8n model for fast and accurate object detection.

**Keywords:** Image data, Computer vision, Object detection, Convolutional neural network, YOLO.

### 3. Introducción

El sector del comercio acuícola en Ecuador enfrenta complicaciones relacionadas con los procesos logísticos automatizados, sobre todo en la creciente demanda internacional y la transición gradual hacia la automatización de las industrias 4.0, limitada por variables como la inversión y la tecnología [1]. Generalmente en las costas del país, específicamente en la parroquia Hualtaco, provincia del Oro, es resaltante la necesidad de la adopción de métodos modernos para el área de logística, como se evidencia en la empresa comercial de productos acuícolas privada ASOCAM (Asociación de Productores Camaroneros) (ver **Sección 5.3.1.1.**), exhibe problemas en los procesos de entrega de los productos, generando problemas en el reconocimiento de empaques de sacos parecidos y el conteo de inventario, utilizando métodos tradicionales como el conteo manual y captura de fotografías a la hora de despachar para tener un comprobante de entrega; sin embargo, esto resulta favorable para una cantidad mínima de productos en los despachos, pero al momento de tener un gran volumen dificultan el control del inventario y ralentizan las entregas.

Por lo que se opta por realizar una entrevista a un maestro en la creación de conjunto de datos de imágenes que puedan ayudar en la automatización de los productos dentro del sector acuícola (ver **Sección 5.3.1.2.**), en donde se deja en claro que los datos pueden ser el punto de partida más importante para la creación de modelos, por lo tanto, un cambio positivo en la gestión operativa en los inventarios; en el presente proyecto se seleccionó la categoría de insumos, tomando tres diferentes productos empaquetados en sacos de polipropileno tejido por su disponibilidad en la empresa ASOCAM y adquisición por parte de los clientes, además de que los repositorios de imágenes del sector acuícola, como en su propia aplicación en general en este sector, no poseen datasets relevantes. Al realizar la búsqueda de un conjunto de datos relacionados en las plataformas de Huggingface, Roboflow, Dataset Search y Kaggle, se encontró una cantidad ínfima de estos, mostrando conjuntos de datos como "Fish Video Object Tracking Dataset", "Water quality judgment" y "Fish detection dataset", siendo estos pobres y ligeros en cuanto a la cantidad de imágenes se refiere, con un promedio de 530 en promedio por conjunto de datos, la creación de un conjunto específico junto con un modelo de inteligencia artificial se plasma como una de las soluciones para mejorar el proceso operativo, reduciendo los tiempos de ejecución en las tareas, la dependencia de los procesos tradicionales y minimizando los errores humanos, que son resultado de las pérdidas económicas y la mala calidad del servicio, con el propósito de realizar un despacho de entrega de productos más rápido y con una menor tasa de error. Ejecutando una visión general a las posibles tecnologías que puedan aportar a la industria comercial, sobresale las tecnologías de inteligencia artificial, como la visión por computadora que se menciona en [2], donde se destaca las últimas versiones de You Only Look Once

(YOLO), para el reconocimiento rápido de objetos, que pueden facilitar los procesos logísticos en general. Por lo que se construye la pregunta de investigación “¿Qué porcentaje de precisión y tasa de error se puede alcanzar en la detección de sacos de balanceado de camarón Aquaxcel GROWER 5 (25 kg), Aquaxcel Pre-Starter 2 (25 kg), y ECOFEED Purina (25 kg) a través de un dataset especializado, utilizando el modelo YOLOv8n?”, con base en esta pregunta surge el objetivo general del proyecto “Determinar el porcentaje de precisión y tasa de error en la detección de sacos de balanceado de camarón Aquaxcel GROWER 5(25 kg), Aquaxcel Pre-Starter 2 (25 kg), y ECOFEED Purina (25 kg) mediante la implementación de un dataset especializado y el modelo YOLOv8n, para establecer un punto de referencia en trabajos futuros de visión por computadora” con el primer objetivo específico “Recolectar imágenes de sacos de balanceado de camarón Aquaxcel GROWER 5 (25 kg), Aquaxcel Pre-Starter 2 (25 kg) y ECOFEED Purina (25 kg) para anotarlas y formar un dataset diverso y representativo, tomando como referencia la metodología CRISP-DM y las prácticas de MLOps” y el segundo objetivo específico “Utilizar el modelo YOLOv8n con el dataset creado para medir la precisión y la tasa de error en la detección de sacos de balanceado de camarón Aquaxcel GROWER 5(25 kg), Aquaxcel Pre-Starter 2 (25 kg), y ECOFEED Purina (25 kg)”.

## **4. Marco teórico**

### **4.1. Antecedentes**

El crecimiento en las últimas décadas de la industria acuícola por la globalización del comercio internacional, ha sobresalido dentro del país, aportando de forma importante a este, atrayendo consigo la competitividad internacional, y su necesidad de automatizar y modernizar los procesos dentro de la industria para mantenerse competitivos [1]; realizando un enfoque más específico dentro de las costas ecuatorianas, se encuentra en la parroquia Hualtaco, provincia del Oro, empresas como ASOCAM, que tiene la necesidad de resolver problemas de eficiencia logística, como la gestión en los inventarios y los procesos en la entrega de productos, que por sus aplicaciones manuales están en un punto de estancamiento en la competitividad (ver **Anexo 2**), lo que trae consigo el incremento de los errores humanos, evidenciando la necesidad de la implementación de sistemas de deep learning para mejorar los procesos operativos, como es el caso de la detección de productos del sector acuícola.

La creación de un conjunto de datos específicos para ciertos problemas que pueden resolverse mediante la visión, por su detección o clasificación, son importantes según trabajos como [23] y [51], que muestran cómo la calidad en su construcción es de mucha importancia, resaltando los datos que son la fuente primaria en este contexto, utilizando la metodología CRISP-DM y las prácticas MLOps como referencia para tomar su marco estructurado para la preparación de los datos y asegurar su reproducibilidad en proyectos similares de machine learning, además de la utilización de versiones de modelos pre entrenados como YOLOv8 que se han utilizado en entornos de producción, mostrando resultados eficientes en aplicaciones de visión por computadora [48] [50]; los repositorios de datos enfocados al sector acuícola tampoco resaltan como tema en las investigaciones, obteniendo como consecuencia, muy pocos conjuntos de datos de calidad para utilizar.

### **4.2. Fundamentación Teórica**

#### **4.2.1. Inteligencia Artificial**

La inteligencia artificial, que en inglés se traduce como “Artificial Intelligence”, es la materia relacionada con la computación, aunque algunos autores filósofos también la relacionan con la complejidad de la mente humana, por su capacidad hoy en día en la toma de decisiones, y también en su capacidad para aprender mediante la experiencia que va adquiriendo por cada interacción [3]; naciendo tales conceptos con Alan Turing, considerado el padre de la computación moderna, que no plasma una definición para lo que hoy se conoce como inteligencia artificial, sino que plantea preguntas tales “¿Cómo una

computadora puede comportarse como una máquina sin que sea distinguible a un ser humano?”, por ello se conoce el popular test de Turing, que inician con la ciencia de tecnologías de imitación humana para aplicaciones muy específicas, con una consideración al acceso a los datos y la aplicación de algoritmos adecuados para poder evitar sesgos en el aprendizaje o entrenamiento de estas tecnologías [4], en donde su aplicación sería inevitable en las industrias, por lo que reduciría el porcentaje de error en las tareas que tradicionalmente están asignadas a los seres humanos [5].

#### **4.2.2. Machine Learning**

Son algoritmos o programas de software que son capaces de identificar patrones o características ya que son entrenados con datos relacionados a un problema en específico ganando experiencia en cada iteración [6].

Son utilizados en la visión por computadora porque su aplicabilidad se ha popularizado mucho con el tiempo como es el caso del reconocimiento facial, que todos los celulares modernos tienen implantado en sus cámaras [7], pero no se queda ahí, también existe para el texto “NPL” (Procesamiento de Lenguaje Natural), reconocimiento de voz, entre otros.

Sus métodos de aprendizaje supervisado, no supervisado y por refuerzo son capaces de realizar predicciones, estos métodos se utilizan en algoritmos K-Nearest Neighbors (KNN) y máquinas de vectores de soporte (SVM) [8].

#### **4.2.3. Deep Learning**

Parte de la materia de Machine Learning realizando un enfoque a las redes neuronales más profundas, la cual se relaciona más al aprendizaje humano que les permite reconocer características y patrones necesarios para poder dar una solución, es aplicada en la visión artificial porque puede identificar objetos en una imagen que cumplen con características específicas [9]

Ha logrado avances significativos para la inteligencia artificial especialmente en la detección de objetos, gracias a las capas convolucionales que ha permitido el desarrollo de los modelos actuales de machine learning, en estos modelos extraen las características de una imagen [9][10][11].

#### **4.2.4. Sistemas Inteligentes**

Con la aceptación actual que recibe la tecnología en el día a día de nuestra sociedad, es imprescindible no utilizarla con la incorporación del campo de la inteligencia

artificial para automatizar actividades que antes solo lo podía cumplir el ser humano, para esto se utiliza los sistemas inteligentes, que gracias a la industria 4.0 se puede realizar trabajos entre humanos y máquinas siendo más eficiente que el trabajo tradicional, estos sistemas tienen ciertas habilidades que a medida que realizan trabajos se van adaptando mediante el aprendizaje, por esto y su continua avance se ha implementado en los campos de agricultura, agroalimentario, automotriz, etc [12][13].

#### **4.2.5. Visión por Computadora**

Es un campo derivado de la inteligencia artificial el cual se basa en herramientas o tecnologías para la percepción de los objetos a través de entradas de tipo imagen o video [14] [15], tiene como propósito entender su entorno mediante la visión, así como lo haría un humano al ver con sus ojos, para reconocer los diferentes tipos de objetos en el cual fue entrenado.

La visión por computadora es la parte de la entrada al sistema, mientras que la IA dentro de ella es la parte que procesa esas entradas para comprenderlas por lo que se puede decir que son como los ojos y los pensamientos humanos, pero a diferencia de las personas esta necesita tener una gran cantidad de diferentes imágenes para que la visión por computadora pueda funcionar correctamente, ya que no puede interpretar las imágenes cómo lo hacen los humanos [16]. Dentro de su procesamiento se reconocen patrones y características que le permiten interpretar las imágenes o videos; actualmente es un área muy importante porque es aplicada en tecnologías del sector primario, como también la seguridad, el desarrollo tecnológico, educación, etc [14] [16].

#### **4.2.6. Fine Tuning**

Es utilizado para pre entrenar redes neuronales profundas debido a su eficiencia en el entrenamiento de modelos multitarea, meta aprendizaje y aprendizaje de disparo cero, pero al utilizarla se puede presentar el problema de sobre ajuste que requiere una medida muy precisa de la generalización de las redes neuronales profundas [17] [18].

Es importante utilizarlo si se desea aprovechar el aprendizaje previo de un modelo ya entrenado, permitiendo ajustar sus pesos a los nuevos datos introducidos durante el siguiente entrenamiento<sup>1</sup>.

#### **4.2.7. Anotación**

Es un proceso realizado en los datos de audio, video, texto o imagen asignados a los modelos de machine learning en el aprendizaje supervisado, donde se puede establecer criterios y características para cada clase, pero son más conocidos en la anotación de

---

<sup>1</sup> [El Fine-Tuning en Aprendizaje Profundo | Las cosas de internet](#)

imagen, que es la que ocupa la visión artificial, este proceso al ser implementado en las imágenes suele ser manual, reconociendo al objeto que se desea etiquetar e indicando el formato en que se desea guardar, esto se repite con todas las imágenes que se tengan a disposición, esto se puede alcanzar con Labelimg la cual facilita el proceso<sup>2 3</sup>, esto se puede hacer de 3 formas<sup>4</sup> :

- **Clasificación:** Etiqueta las imágenes según su contenido y le asigna una clase, pero puede representar un problema en situaciones donde las imágenes contienen varios objetos o características complejas.
- **Detección de Objetos:** Localiza al objeto dentro de la imagen y le asigna la clase lo que le permite encontrar varios objetos en una sola imagen. Lo cual se puede lograr con técnicas como los cuadros delimitadores, que mediante rectángulos o cuadrados encierran al objeto en la posición que se encuentre; Segmentación poligonal, el cual requiere el uso de polígonos para identificar objetos dentro de la imagen, ideal para objetos asimétricos donde no es recomendable usar cuadros; y Líneas, que es una técnica que utiliza líneas para imágenes muy largas, como por ejemplo las vías de un ferrocarril.
- **Segmentación Semántica:** Encierra cada clase por separado ayudando a los modelos a reconocer elementos que podrían estar ocultos.

#### 4.2.8. Detección de Objetos

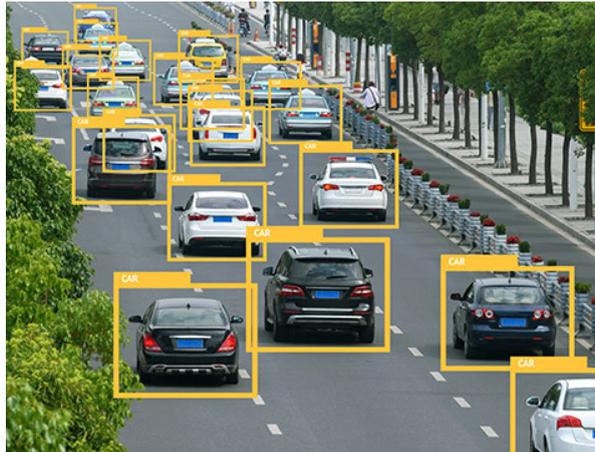
Con la aparición de la visión artificial se busca que las computadoras fueran capaces de reconocer objetos de manera rápida y precisa, para esto se utilizaron clasificadores de imágenes, pero presentaban problemas para identificar múltiples objetos en una sola imagen, debido a esto aparecieron los algoritmos de detección de objetos que tienen la capacidad de localizar objetos dentro de la imagen o video [19][20]. Las ubicaciones de los objetos pueden ser diferentes dentro del rango de visión del dispositivo por ello se utiliza los algoritmos YOLO, RNN o DNN para reconocer patrones en la imagen que pertenecen una clase y encerrarla utilizando cuadros delimitadores demostrando una buena precisión [21] (ver **Figura 1**).

---

<sup>2</sup> [Anotación de datos: mejores herramientas, desafíos y beneficios en 2024 \(shaip.com\)](https://shaip.com)

<sup>3</sup> [ETIQUETADO / ANOTACIÓN DEL CONJUNTO DE DATOS DE IMAGEN \(ichi.pro\)](https://ichi.pro)

<sup>4</sup> [Técnicas de anotación de imágenes con implementación en OpenCV \(pctq.net\)](https://pctq.net)



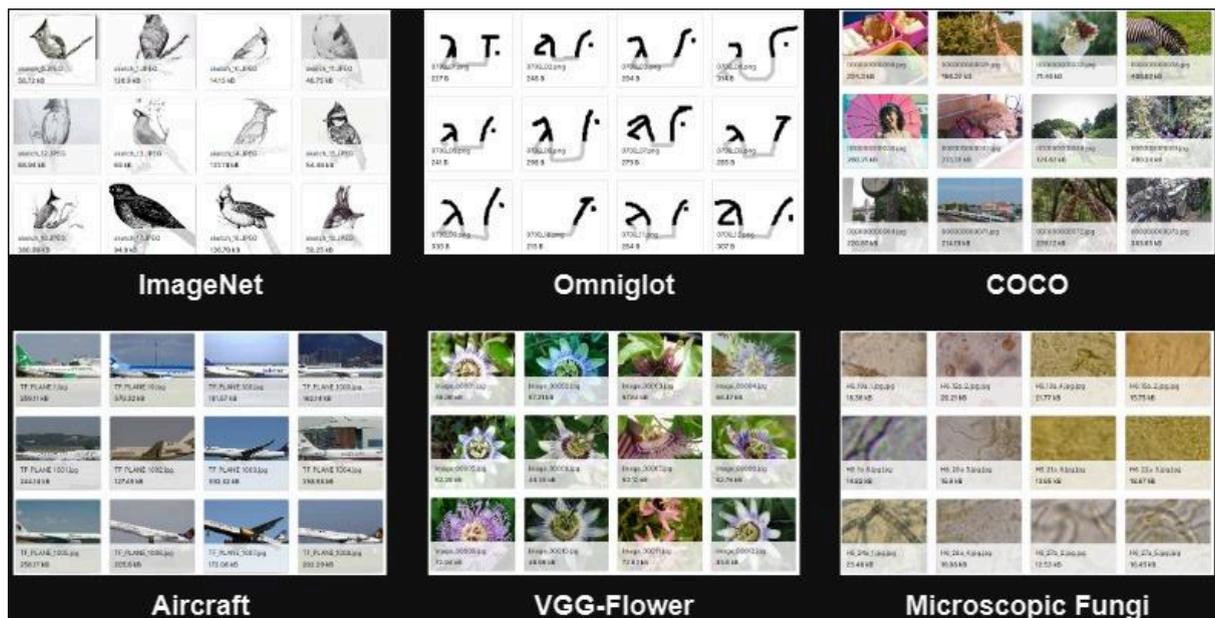
**Figura 1.** Aplicación de la detección de objeto para una cámara vigilancia para que capte carros (Car) [22].

#### 4.2.9. Conjunto de Datos (Dataset)

Son importantes para entrenar modelos machine learning en el campo de la inteligencia artificial. Es importante que el conjunto de datos sea variado, con una gran cantidad de datos, dependiendo del caso, para asegurar un entrenamiento efectivo y preciso del modelo.

En el caso de un conjunto de datos para la detección de objetos, es necesario utilizar técnicas supervisadas para el procesamiento y la realización de etiquetas que rodean el objeto de interés [19], la obtención de un dataset es importante para investigaciones de los modelos, porque se toman en cuenta las características y etiquetas que están aprendiendo, para resolver problemas específicos de visión por computadora [10] [23].

Son el punto de partida para los porque desde ellos se puede realizar el entrenamiento, la evaluación y las pruebas para comprobar que el modelo esté funcionando correctamente, además de poder entrenar distintos modelos con los mismos datos para comparar sus métricas, es necesario que un dataset este correctamente desarrollado para evitar sesgos, tecnología defectuosa y resultados imprevistos [23]. La **Figura 2** se muestran algunos ejemplos de dataset populares actualmente.



*Figura 2. Ejemplos de diferentes conjuntos de imágenes populares actualmente, realizada por el autor del presente TIC.*

#### 4.2.10. Sacos de Balanceado de camarón

Las empresas del sector acuícola en Ecuador son muy importantes para el crecimiento económico del país, estableciendo la tarea de criar en piscinas diversos tipos de animales acuáticos, destacando principalmente el camarón. Por esta razón, es de necesidad tener los insumos necesarios para una buena crianza y producción eficientes.

Es necesario agilizar algunos procesos que tradicionalmente lo hacen los humanos, como los de gestión logística. Para poder adoptar tales tecnologías a los sistemas inteligentes es necesario conocer los productos como Aquaxcel GROWER 5 (25 kg), Aquaxcel Pre-Starter 2 (25 kg) y ECOFEED Purina (25 kg). A continuación los detalles de cada uno de ellos:

- **Aquaxcel GROWER 5 (25 kg):** Alimento de balanceado extruido diseñado específicamente para el camarón blanco del Pacífico (*Litopenaeus vannamei*) en su etapa de crecimiento, este producto mejora la digestibilidad y la absorción de nutrientes, promoviendo un ambiente de cultivo saludable y sostenible<sup>5</sup>.
- **Aquaxcel Pre-Starter 2 (25 kg):** Es un alimento balanceado extruido funcional, diseñado para satisfacer los requerimientos nutricionales del camarón blanco (*Penaeus vannamei*) en cultivo. Está formulado para la etapa de post-larva en

<sup>5</sup> [Aquaxcel-GROWER](#)

sistemas de pre-cría o raceways. Su composición incluye un 42% de proteína, lo cual es crucial para el crecimiento y desarrollo temprano del camarón<sup>6</sup>.

- **ECOFEED Purina (25 kg):** Trata de un alimento balanceado peletizado diseñado para camarones, especialmente formulado para proporcionar resultados nutricionales que incluye un equilibrio adecuado de proteínas, grasas y otros nutrientes esenciales. Se enfoca en optimizar el crecimiento y la salud de los camarones durante su cultivo, fabricado por Cargill<sup>7</sup>.

### 4.3. Metodologías y Prácticas en Ciencia de Datos y Machine Learning

#### 4.3.1. Metodología CRISP-DM

La metodología CRISP-DM (Cross Industry Standard Process for Data Mining) por sus iniciales en español significa “Proceso Estándar Inter-Industrias para Minería de Datos” es principalmente promovida por la empresa IBM y es utilizada para proyectos de minería de datos siendo muy flexible para los trabajos que estén relacionados con los datos y un estándar del mismo campo a nivel mundial [24].

Esta metodología contiene seis fases en total como se muestra en la **Figura 3** y no es un proceso de una sola dirección, sino que es bidireccional, lo que quiere decir que se puede regresar a una fase anterior si es necesario [25]. Sus fases se componen por [24] [25] [26]:

- **Comprensión del Problema:** Esta fase define los objetivos y criterios del problema, para tener claro lo que se desea lograr a lo largo del proceso, para el cumplimiento de los objetivos.
- **Comprensión de datos:** En esta siguiente fase se realiza una comprensión de los datos que se vayan a utilizar, para entender que datos y características clave recolectar y qué técnicas de visualización optar para el proyecto.
- **Preparación de datos:** Aquí es donde los datos son revisados, por lo tanto, también es una fase que requiere su tiempo, para que se procesen y se le dé una estructura, además de ser limpiados, y transformados si es necesario.
- **Modelado:** Se selecciona el mejor modelo.
- **Evaluación de Modelo:** Se evalúa al modelo elegido con métricas para verificar su fiabilidad, esta evaluación también puede ser realizada con el apoyo de un experto para ver si cumple con los objetivos establecidos.

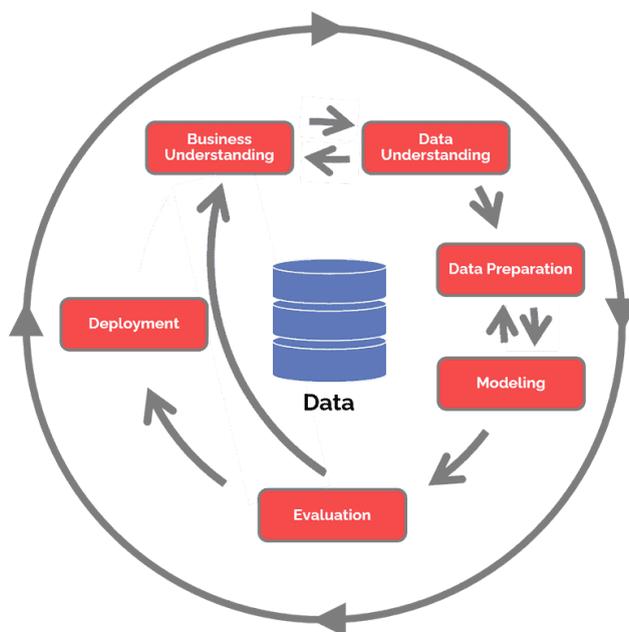
---

<sup>6</sup> [Aquaxcel-Pre-Starter](#)

<sup>7</sup> [ECOFEED-shrimp](#)

- **Implementación de Modelo:** Se monitorea al modelo y se revisa su rendimiento.

A pesar de que esta metodología está pensada para la minería de datos, se la puede adaptar y/o personalizar a necesidades más específicas<sup>8</sup>, como la creación de un conjunto de datos para machine learning.



*Figura 3. Diagrama de flujo de la metodología CRISP-DM [25].*

#### 4.3.2. Prácticas de MLOps

Las prácticas de MLOps ayudan a mitigar este riesgo al acelerar el desarrollo, despliegue y mantenimiento de los modelos de machine learning [27], compuesta en parte de la metodología DevOps (ver **Figura 4**), tiene como objetivo desplegar y mantener a estos modelos con más eficiencia y rapidez. Su enfoque multidisciplinario ayuda a cumplir con los procesos relacionados con los modelos de machine learning utilizando las siguientes fases<sup>9</sup> [27] [28] [29]:

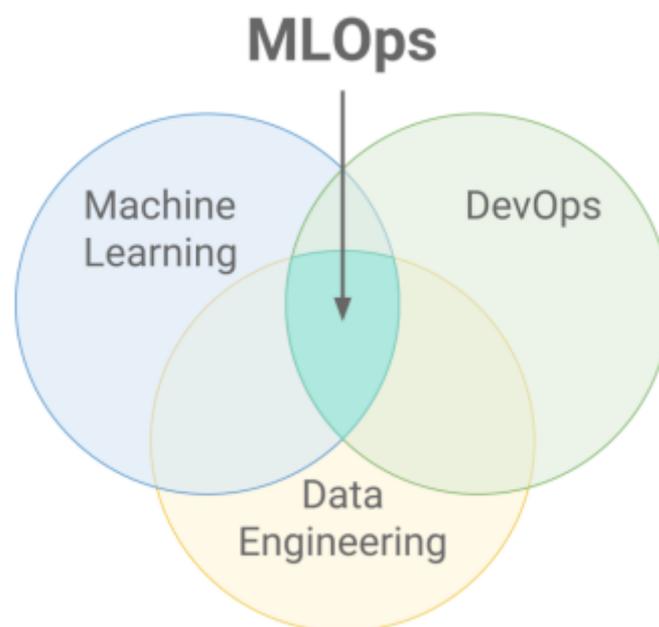
- **Crear:** Se crea al modelo junto a su preparación y características tomando en cuenta la ingeniería de datos.
- **Administrar:** Se realiza el control de versiones al modelo creado.
- **Implementar:** Se implementa al modelo en el sistema inteligente.

<sup>8</sup> [What is CRISP DM? - Data Science Process Alliance \(datascience-pm.com\)](https://datascience-pm.com/)

<sup>9</sup> [MLOps Principles \(ml-ops.org\)](https://ml-ops.org/)

- **Monitorear:** El modelo va a estar en constante monitoreo para poder evaluar su rendimiento y determinar si es óptimo, para que los nuevos datos que se obtengan puedan darle evolución al modelo de machine learning.

A pesar de que está dirigido para el desarrollo y despliegue de los modelos de machine learning como se puede evidenciar con las fases anteriormente mencionadas, se puede considerar algunas prácticas iniciales para el desarrollo de un conjunto de datos, para que cumpla con lo necesario que necesite un conjunto de datos y sea implementado a los modelos de machine learning.



*Figura 4. Elementos que consideran las prácticas de MLOps<sup>10</sup>.*

#### 4.4. Tecnologías

##### 4.4.1. Google Drive

Es una herramienta gratuita que ofrece 15 GB de almacenamiento seguro en la nube, ofrece servicios fáciles de usar y planes con mayor capacidad que pueden ayudar para proyectos más pesados que requieran más espacio de almacenamiento gracias a eso se puede trabajar con proyectos en Colab [31]. Esta tecnología es necesaria para trabajos que requieran guardar versiones de datos, por ser un repositorio con un almacenamiento limitado dependiendo del plan, para poder compartir cualquier tipo de archivo [32]. Para

---

<sup>10</sup> [Databitai.com/machine-learning/mlops/](https://databitai.com/machine-learning/mlops/)

poder tener acceso a los beneficios que ofrece Drive, se tiene que tener una cuenta de correo electrónico de Gmail [30].

#### **4.4.2. Google Colaboratory**

Es una alternativa gratuita para realizar procesamiento y entrenamiento de modelos de machine learning de forma local obteniendo resultados en menor tiempo [33], permite escribir código en Python facilitando el desarrollo de proyecto de programación de robots y la ciencia de datos [11][31], además ofrece servicios de GPU o CPU en la nube de forma gratuita para ser utilizados pero con limitaciones de recursos, aunque también prevé planes para no tener estos inconvenientes [33].

#### **4.4.3. Labellmg**

Es una herramienta muy sencilla de usar que permite realizar el etiquetado en imágenes utilizadas en el entrenamiento de modelos de machine learning para la detección de objetos pero solo en cuadros o rectángulos delimitadores, fue creada por Tzutalin y escrita en Python, guarda sus salidas en formatos como PASCAL VOC (Pascal Visual Object Classes), YOLO y CreateML [9][34].

El etiquetado con esta herramienta se realiza de forma manual y supervisada, lo que permite identificar el objeto en cuestión dentro de la imagen y capturarlo en un cuadro delimitador, como se evidencia en los trabajos [54], [55] [56].

#### **4.4.4. Visual Studio Code**

Es un excelente editor de código que ofrece un extenso ecosistema de extensiones para diversos lenguajes de programación, se ejecuta de forma local y es compatible con sistemas operativos como Linux, macOS y Windows [35][36].

#### **4.4.5. Github**

Utilizada para el acceso a los datasets o códigos relacionados por el alojamiento de código y el control de versiones, aumentando la competitividad en proyectos de código abierto Aunque la plataforma facilita el acceso a códigos y datos abiertos, es importante tener en cuenta las licencias asociadas [37][38][39].

#### 4.4.6. Python

Utilizado en proyectos de ciencia de los datos debido a que es fácil de utilizar, tiene actualizaciones cada cierto tiempo que mejoran su comprensión y su utilidad [40]. Proporciona un vasto conjunto de herramientas y librerías que lo hacen popular, además de ser el principal lenguaje de programación para desarrollar tecnologías de Machine learning, como NLP (Natural Language Processing), Visión por computadora y texto [41].

Es utilizado para el despliegue de modelos de machine learning. Su naturaleza interpretada y orientada a objetos, además de ser de libre uso, lo convierte en una herramienta muy versátil [40], facilita la descarga de datasets permitiendo su división en subconjuntos de entrenamiento (train), validación (val) y prueba (test), automatizando así los procesos. Además, proporciona métricas que permiten visualizar el rendimiento de los modelos [41].

#### 4.4.7. OpenCV

Es utilizada en machine learning para el campo de visión por computadora y el procesamiento de imágenes lo convierte en una herramienta fundamental en el aprendizaje de este campo, se maneja en múltiples plataformas y lenguajes popularmente en Python, tiene licencia BSD (Berkeley Software Distribution), lo que permite su uso comercial [42]. Es especialmente útil para la detección, reconocimiento y segmentación de objetos [43]. Y con el creciente interés en los últimos años por la inteligencia artificial, es más acogida para pruebas y aplicaciones industriales [44].

### 4.5. Modelo de Red Convolutiva para Detección de Objetos

#### 4.5.1. YOLOv8n

El modelo YOLOv8n parte de YOLO (You Only Look Once) es un framework utilizado para los proyectos de visión por computadora en clasificación, segmentación y la detección de objetos ideal para entrenamientos de detección de objetos con características similares, se han sacado nuevas versiones cada vez mejores, siendo una de ellas YOLOv8 su arquitectura está basada en YOLOv5 aprovechando las fortalezas de sus predecesoras (ver **Figura 5**), tiene sus modelos como YOLOv8n, YOLOv8s, YOLOv8m, YOLOv8l y YOLOv8x, cada una con diferentes requerimientos al igual que resultados siendo YOLOv8n es la más ligera de ellas debido a su velocidad y precisión al realizar predicciones en tiempo real[45] [46].



Código	Título	Resumen
		específicas.
TR02	Improved Fire Detection by YOLOv8 and YOLOv5 to Enhance Fire Safety	Este documento propone un marco de detección de incendios utilizando el algoritmo YOLO, evaluando las versiones YOLOv5 y YOLOv8, incluyendo tres submodelos de cada uno (Nano, Medium y Large), entrenados en un conjunto de datos sobre incendios y en un conjunto de datos personalizado propio. Analizando seis modelos YOLO para examinar su eficacia y compararlos con otros sistemas existentes para la detección de incendios basados en CNN, YOLOv8 obtuvo una precisión del 99.1% en el conjunto de datos de referencia y 94.5% en el conjunto de datos personalizado [49]. Esta alta tasa de precisión destaca el potencial de integrar estos modelos en sistemas modernos.
TR03	A Small Target Strawberry Recognition Method Based on Improved YOLOv8n Model	Este estudio utiliza el modelo YOLOv8n para resolver problemas de detección insuficiente de fresas pequeñas y errores de detección debido a fondos complejos en imágenes de fresas. Al implementar un conjunto de datos creado obtuvieron una reducción del volumen de un 59.7%, la precisión aumentó en un 1.3%, la tasa de recuperación en un 2.1%, el mAP mejoró en un 1.6% y el tiempo de detección por imagen de fresa en una GPU fue de 17.2 ms demostrando un buen rendimiento de detección [50].
TR04	A construction waste landfill dataset of two districts in Beijing, China from high resolution satellite images	Este estudio desarrolla un conjunto de datos específico de vertederos de residuos de construcción en los distritos. El conjunto de datos comprende 3,653 muestras de áreas de imágenes originales y proporciona imágenes con etiquetas de máscara para la segmentación semántica. Cada píxel dentro de un vertedero de residuos de construcción se clasifica en cuatro categorías: área de fondo, área vacante rellenable, área de instalaciones de ingeniería y área de vertido de residuos. El conjunto contiene 237,115,531 píxeles de residuos de construcción y 49,724,513 píxeles de instalaciones de ingeniería. Las etiquetas de segmentación semántica a nivel de píxel se proporcionan para cuantificar la producción de residuos de construcción, lo que sirve como datos básicos para la extracción y estimación del rendimiento de residuos de construcción, tanto para investigación académica como industrial [51].
TR05	A dataset for fine-grained seed recognition	Este estudio aborda la falta de conjuntos de datos en el campo agrícola para el entrenamiento de modelos de visión por computadora. Se utilizó un dispositivo basado en teléfonos móviles y lentes macro para adquirir las imágenes obteniendo 4496 imágenes de 88 tipos diferentes de semillas sirviendo para entrenar modelos de aprendizaje profundo en el campo informático y proporcionando un apoyo de datos crucial para la investigación agrícola y forestal [52].

Código	Título	Resumen
TR06	Deep learning-based image annotation for leukocyte segmentation and classification of blood cell morphology	En este estudio se segmentan y clasifican leucocitos mediante técnicas avanzadas de procesamiento de imagen y redes neuronales convolucionales (CNN). Utiliza un conjunto de datos de 12,444 imágenes de leucocitos divididas en cuatro clases: monocitos, linfocitos, eosinófilos y neutrófilos. El preprocesamiento incluye la eliminación de ruido y la creación de máscaras de fondo y leucocitos, seguido de técnicas de aumento de datos para mejorar la generalización del modelo. Los resultados muestran una precisión del 97.98% y una precisión del 97.97%, demostrando la eficacia del modelo en la segmentación y clasificación precisa de leucocitos, lo que contribuye significativamente al diagnóstico médico [53].
TR07	E-YOLO: Recognition of estrus cow based on improved YOLOv8n model	Este estudio utiliza Estrus-YOLO que es una versión mejorada de YOLOv8n, para identificar vacas en celo de manera precisa y rápida. Integraron módulos de Atención Triplet y de Aumento de Información Contextual para mejorar las características relevantes. Obtuvieron una Precisión Promedio de celo (APestrus) del 93.90% y una velocidad de detección de 8.1 ms por cuadro, lo que lo hace ideal para el monitoreo en tiempo real [47].
TR08	Aplicación de técnicas de aprendizaje profundo (deep learning) para la detección de objetos en Industria 4.0	Este trabajo explora la aplicación de técnicas de aprendizaje profundo en la Industria 4.0, enfocándose en el control de calidad mejorado a través de la visión artificial. Desarrollan un clasificador de imágenes para detectar impulsores de bombas defectuosas logrando una precisión del 99.24% con la mejor configuración y un detector de objetos para identificar defectos en placas de circuitos impresos (PCBs) con el modelo RetinaNet que alcanzó un mAP del 93.78% en la detección de defectos en PCBs, utilizando un conjunto de datos etiquetado en formato PASCAL VOC [19].
TR09	Data and its (dis)contents: A survey of dataset development and use in machine learning research	Este trabajo es una revisión exhaustiva de la literatura sobre las limitaciones de las prácticas predominantes en la recolección y uso de datos en el campo del aprendizaje automático. Se abordan estudios que revisan críticamente el diseño y desarrollo de conjuntos de datos, enfocándose en los impactos negativos para la sociedad y los malos resultados en el rendimiento de los sistemas, discuten técnicas de filtrado y aumento de datos, así como técnicas de modelado destinadas a mitigar el impacto del sesgo en los conjuntos de datos, se examinan las prácticas, culturas y normas disciplinarias en torno a los datos, discutiendo las implicaciones legales, éticas y funcionales que el campo sigue enfrentando. A partir de estos se aboga por el uso combinado de enfoques cualitativos y cuantitativos analizando cuidadosamente el conjunto de datos durante las fases de creación y uso [23].

## 5. Metodología

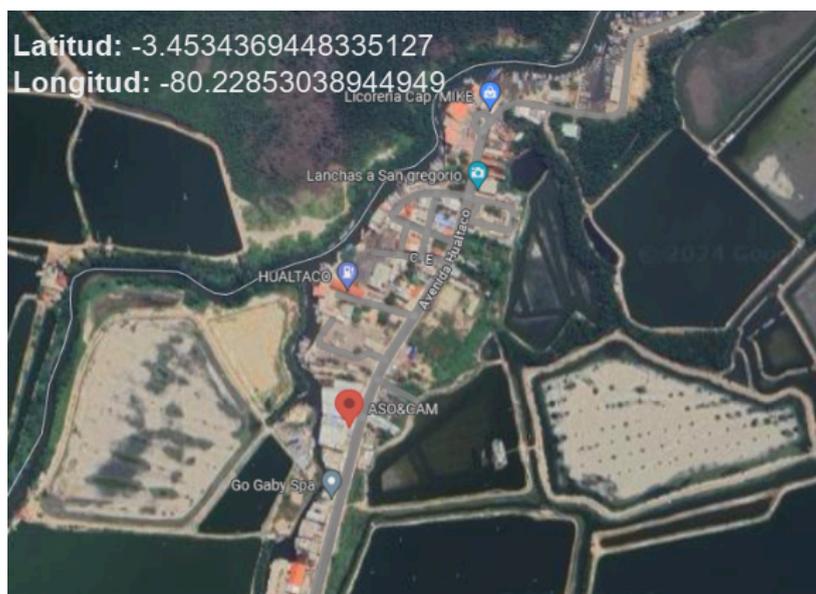
El presente trabajo de integración curricular (TIC) con el tema “Desarrollo de un Dataset Especializado para la Detección Automática de Sacos de Balanceado de Camarón Aquaxcel GROWER 5 (25 kg), Aquaxcel Pre-Starter 2 (25 kg) y ECOFEED Purina (25 kg) utilizando el Modelo YOLOv8n” se centró en la creación de un conjunto de datos de imágenes con etiquetas en cuadros delimitadores, aplicando técnicas de detección de objetos. Estos datos se utilizaron para entrenar y analizar el modelo YOLOv8n, enfocado en la detección de sacos de balanceado de camarón comercializados en empresas acuícolas.

En las siguientes secciones se describen los procedimientos y materiales utilizados en el TIC.

### 5.1. Área de estudio

La recolección de datos se realizó en la parroquia Hualtaco, del cantón Huaquillas, provincia del Oro, Ecuador, en la empresa privada ASOCAM, como se muestra su ubicación en la **Figura 6**.

ASOCAM ofrece servicios a medianos y pequeños productores camaroneros, enfocándose en prácticas sostenibles que respeten el medio ambiente.



**Figura 6.** Mapa de la ubicación de ASOCAM en la parroquia Hualtaco, provincia del Oro, Ecuador.

## 5.2. Procedimiento

A continuación, se desglosa las tareas que se ejecutaron para el cumplimiento de los objetivos establecidos:

**5.2.1. Objetivo 1: Recolectar imágenes de sacos de balanceado de camarón Aquaxcel GROWER 5 (25 kg), Aquaxcel Pre-Starter 2 (25 kg) y ECOFEED Purina (25 kg) para anotarlas y formar un dataset diverso y representativo, tomando como referencia la metodología CRISP-DM y las prácticas de MLOps.**

Para abarcar el primer objetivo se toma en cuenta la entrevista realizada al experto en dataset, el Ing. Oscar Miguel Cumbicus Pineda como se puede evidenciar en la **Sección 5.3.1.2.**, que actualmente es docente en la Facultad de la Energía, las Industrias y los Recursos Naturales No Renovables de la Universidad Nacional de Loja (UNL), donde se menciona como respuesta en la primera pregunta “si no hay datos no se pueden hacer ni los modelos de inteligencia artificial o de machine learning, por lo tanto, lo importante o la más alta importancia son los datos, en este caso los datasets” y se tomó aspectos clave en la creación de un dataset para la visión por computador. Con base en esta entrevista, las buenas prácticas MLOps (ver **Sección 4.3.2.**) y tomando como referencia a la metodología de minería de datos CRISP-DM, se definieron las siguientes tareas:

### 5.2.1.1. Tarea 1. Captura de datos.

Se inició tomando la primera fase de CRISP-DM, que trata sobre la comprensión de datos (Data Understanding), que se aplicó para ver el estado actual de la empresa ASOCAM con la ayuda del recurso técnico de la entrevista (ver **Sección 5.3.1.1.**), se realizó a la Ing. Marjorie Aponte Suárez responsable de administración y contabilidad, donde se mencionan las dificultades de la gestión y del control logístico de las instalaciones de la empresa. Esta interacción reveló varias problemáticas importantes, destacando la falta de un conjunto de datos para la implementación de sistemas inteligentes, obligando a los empleados a tomar medidas manuales, por la falta de tecnologías que faciliten estos procesos, para que ayuden a una solución eficiente y con menor tasa de error en las respuestas. Con la autorización de la empresa, firmada por Marjorie Suarez (ver **Anexo 3**), se inició con la parte práctica del proyecto, realizándose en las instalaciones de ASOCAM (ver **Figura 7**), tomando los escenarios (a), (b), (c), y (d) para darle diferentes entornos e iluminación a los datos. Eligiendo los sacos de balanceado para camarón, que son Aquaxcel GROWER 5 (25 kg), Aquaxcel Pre-Starter 2 (25 kg), y ECOFEED Purina (25 kg) como las clases del conjunto de datos del presente proyecto, se procede a la captura de imágenes, que se llevó a cabo entre las 09h00 y 17h00 del 05 de febrero de 2024, esto se detalla en la **Tabla 2**.



(a)



(b)



(c)



(d)

**Figura 7.** Parte de las instalaciones de la empresa ASOCAM. (a) Vista en dirección a la oficina de Ventas. (b) Vista en dirección al bar y productos acuícolas. (c) Vista en dirección a la entrada a la empresa y salida de despacho vía terrestre. (d) Vista en dirección a la salida de despacho vía marítima.

**Tabla 2.** Recursos utilizados para la recolección de datos.

Recurso 1					
Celular	Modelo de la cámara	Tamaño en Megapíxeles	Apertura de la Cámara	RAM	Procesador
Xiaomi Redmi Note 11	2201117TG	50	f/1.8	4/6 GB	Snapdragon 680
Recurso 2					
Nombre del Saco	Material del Saco	Peso en Kilogramos	Colores Predominantes		
Aquaxcel GROWER 5	Polipropileno tejido	25	Blanco con Azul		
Aquaxcel Pre-Starter 2	Polipropileno tejido	25	Blanco con Rosado		
ECOFEEED Purina	Polipropileno tejido	25	Blanco con Verde		

## Condiciones de Captura

Para asegurar una alta calidad de las imágenes, se establecieron las siguientes condiciones de captura en la **Tabla 3**.

**Tabla 3.** *Condiciones tomadas para la realización de la captura de datos.*

Condiciones de Captura	
<b>Diferentes condiciones de iluminación</b>	Capturar imágenes bajo luz natural, en contraluz, y en áreas con sombra.
<b>Diversos entornos dentro de las instalaciones de ASOCAM</b>	Incluir áreas como el almacén, la zona de carga, y las áreas de producción.
<b>Fotografías con el objetivo cercano</b>	Asegurarse de que el saco esté a una distancia no mayor a 5 metros de la cámara.
<b>Diferentes ángulos de posición</b>	Incluir tomas desde diferentes perspectivas como frontal, lateral y superior.
<b>Captura de imágenes de sacos apilados</b>	Capturar las imágenes cuando estén organizadas en palet de madera.

Se utilizó el modo de captura “ráfaga” de la cámara, con un intervalo de un segundo entre cada toma, capturando hasta aproximadamente 50 imágenes por escena, realizando un movimiento pequeño con cada toma, para obtener imágenes diferentes con cada captura, consiguiendo 3280 imágenes en total.

Se aplicó las prácticas MLOps para el control de versión del conjunto de datos en Google Drive.

### 5.2.1.2. Tarea 2. Preparación de los datos.

Basado en la segunda fase de CRISP-DM, que es la preparación de los datos (Data Preparation), esta sección se enfoca en procesar los datos mediante la limpieza, organización y modificación. Las prácticas MLOps se aplicaron para asegurar el control de versiones en código (En el repositorio GitHub y Google Colaboratory) y en datos (En el repositorio Google Drive) en cada paso del proceso que se llevó a cabo con los datos.

## Clasificación y Separación de Datos

Los datos recolectados se clasificaron y separaron en tres clases: Aquaxcel GROWER 5 (25 kg), Aquaxcel Pre-Starter 2 (25 kg) y ECOFEED Purina (25 kg). Se

eliminaron las imágenes que no pertenecían a ninguna de las clases definidas. Poseyendo 3276 datos en este punto.

### Criterios de Inclusión y Exclusión

Se establecieron criterios de inclusión y exclusión (ver **Tabla 4**) para determinar qué imágenes pasarían a la siguiente fase. Estos criterios se basan en la **Tabla 3**, que se enfoca en tener datos de calidad en condiciones decentes de iluminación y detalles en los objetos, para el procesado de datos.

**Tabla 4.** *Criterios de inclusión y exclusión para los datos recolectados.*

Criterios de Inclusión	Criterios de Exclusión
<ul style="list-style-type: none"><li>• Imágenes que muestran a los sacos de balanceado de camarón en condiciones aceptables de iluminación (Que se noten los detalles).</li><li>• Imágenes muestran a los sacos de balanceado de camarón al menos en un 75% de su cuerpo.</li><li>• Imágenes donde se pueden distinguir al menos un 70% de las características del saco objetivo.</li><li>• Imágenes donde el saco objetivo no se encuentre demasiado lejos desde el punto donde se tomó la fotografía.</li><li>• Imágenes de los sacos objetivos que tengan el recubrimiento de plástico encima y que aún puedan observarse los detalles del saco.</li><li>• Imágenes de los sacos objetivos cuando estén apilados y que se pueda ver al menos una parte de sus caras que contengan una característica representativa.</li></ul>	<ul style="list-style-type: none"><li>• Imágenes borrosas, oscuras o con sobreexposición al punto de no poder notar los detalles del saco objetivo.</li><li>• Imágenes donde los sacos están obstruidos por otros objetos, tapando más del 25% del saco.</li></ul>

### Limpieza de Datos

A continuación, se procedió a la limpieza de los datos basándose en los criterios de inclusión y exclusión mediante la toma de los mismos, revisando de forma supervisada que cada imagen cumpliera con los criterios establecidos (ver **Tabla 3**), y los datos que no cumplieron se procedió a eliminarlos, quedando 3145 datos en total.

## Redimensionamiento de Imágenes

Las imágenes resultantes de la limpieza fueron redimensionadas acorde a los requisitos del modelo YOLOv8n (ver **Tabla 5**). Este proceso se realizó utilizando Visual Studio Code y Python, ajustando las dimensiones a 640 x 640 píxeles.

**Tabla 5.** Requisitos de los modelos de YOLOv8, probados con el dataset COCO.

Model	size (pixels)	$mAP^{val}$ 50-95	Speed CPU ONNX (ms)	Speed A100 TensorRT (ms)	params (M)	FLOPs (B)
YOLOv8n	640	37.3	80.4	0.99	3.2	8.7
YOLOv8s	640	44.9	128.4	1.20	11.2	28.6
YOLOv8m	640	50.2	234.7	1.83	25.9	78.9
YOLOv8l	640	52.9	375.2	2.39	43.7	165.2
YOLOv8x	640	53.9	479.1	3.53	68.2	257.8

## Renombramiento de Imágenes

Para facilitar la organización y el control de versiones, se renombraron las imágenes en una secuencia numérica. Este proceso se llevó a cabo utilizando Google Colab, donde las imágenes de cada clase se numeran de forma ascendente, específicamente como "Imagen\_x" donde la "x" representa un número, comenzando desde el 1.

### 5.2.1.3. Tarea 3. Etiquetado de datos.

Se utilizó la herramienta LabelImg que ofrece una interfaz de usuario que facilita la anotación, para la detección de objetos, configurando el formato de salida de anotación "YOLO" que pertenece a la extensión ".txt", al implementar MLOps para el control de versiones después de la anotación se puede asegurar un seguimiento organizado de los cambios realizados en los datos en Google Drive.

1. En la herramienta se definen las clases que anteriormente son declaradas en la clasificación y separación de la **Tarea 2**, que corresponden a: "Aquaxcel GROWER 5 (25 kg)", "Aquaxcel Pre-Starter 2 (25 kg)", y "ECOFEED Purina (25 kg)".

2. Se aplicaron los criterios de inclusión y exclusión definidos en la **Tabla 4** para asegurar que cada objeto dentro de una imagen con los cuadros delimitadores de la anotación cumpliera con los estándares establecidos.
3. La herramienta LabelImg fue de utilidad para la creación de los cuadros delimitadores a los objetos de interés supervisado según el criterio del autor de este documento.

#### **5.2.1.4. Tarea 4. División de los datos.**

Continuando con la segunda fase de CRISP-DM, se dividió el conjunto de datos, en específico sobre la proporción de los datos, según los trabajos relacionados (ver **Tabla 1**) y las prácticas MLOps para el control de versiones y la división de los datos en entrenamiento (80%), validación (10%) y prueba (10%), obteniendo un proceso mejor organizado tanto en código (En Google Colaboratory) y en datos (En Google Drive), para esto se realizó lo siguiente:

1. Se utilizó Google Colab y código en Python para la división de los datos.
2. Creación del código para la división el conjunto de datos en tres subconjuntos: entrenamiento (train 80%), validación (val 10%) y prueba (test 10%), a su vez estableciendo carpetas como “images” (Imágenes) y “labels” (Etiquetas) para cada subconjunto de datos, para tener un orden en los datos.
3. Luego en el código en google colab se calculó el número total de imágenes y se determinaron los índices correspondientes para cada subconjunto de forma aleatoria.
4. Para que posteriormente las imágenes se distribuyeran en sus respectivas carpetas, copiando también sus archivos de etiqueta.

Obteniendo 2516 en los datos de entrenamiento, 314 en datos de validación y 315 en prueba.

#### **5.2.1.5. Tarea 5. Entrenamiento con el modelo YOLOv8n.**

YOLOv8n es conocido por su capacidad para realizar detección en tiempo real con alta precisión y velocidad, considerando que es el modelo de su versión más ligera, lo que lo hace ideal para aplicaciones donde la rapidez y la exactitud son cruciales, como es el caso de un entorno industrial.

Se utilizó la fase del modelado CRISP-DM para la utilización del modelo YOLOv8n para la configuración a utilizar, que corresponde a la preparación de los hiperparámetros para el entrenamiento.

### 1. Configuración de Hiperparámetros:

- Se utilizó Google Colab, para la configuración de los hiperparámetros del modelo YOLOv8n tomando su propia documentación<sup>12</sup>. Esto incluye ajustes como task (tarea que realizará el entrenamiento), mode, el número de épocas de entrenamiento, data (ruta del archivo yaml, que contiene las rutas de los subconjuntos de datos y los nombres de las clases), model (Modelo elegido para el entrenamiento), imgsz (Tamaño de entrada de los datos del entrenamiento) y batch.

### 2. Ejecución del Ciclo de Entrenamiento:

- a. Se utilizó Google Colab para entrenar el modelo de YOLOv8.
- b. El entrenamiento se realizó utilizando el conjunto de datos de entrenamiento dividido en la **Tarea 4** y se validó con el conjunto de datos de validación.

Se implementaron prácticas MLOps para garantizar la calidad y seguimiento del modelo, lo que compone el guardado del control de versiones, tanto en código como en la mejor versión del modelo entrenado (En Google Colaboratory) y el monitoreo de métricas de precisión y tasa de error en la detección de objetos.

#### 5.2.1.6. Tarea 6. Mejora del Modelo.

En la última tarea del primer objetivo, se utilizó la matriz de confusión junto con las instancias totales de cada clase para mejorar el modelo, sacado de los resultados del entrenamiento y validación de la **Tarea 5**:

1. Se identificaron los problemas para el desequilibrio que mostraban la matriz de confusión y las instancias de cada clase. Donde se muestra una mayor cantidad de instancias a la clase Aquaxcel GROWER 5 (25 kg), que tiene aproximadamente 1000 cuadros delimitadores más que las clases compañeras.
2. Se separó el conjunto de datos por las diferentes clases existentes.
3. Se eliminaron 70 imágenes junto a su etiqueta de la clase con mayor diferencia en número de cuadros delimitadores totales.

---

<sup>12</sup> [YOLOv8 - Ultralytics YOLO Documentos](#)

4. Se renombraron las imágenes junto con su respectiva etiqueta, modificando el código de renombramiento de la **Tarea 2**.
5. Se utilizó el código de la división de los datos de la **Tarea 4**, con los subconjuntos de train (2460 datos), val (307 datos) y test (308 datos), teniendo en total 3070 datos en el nuevo conjunto de datos.
6. En el entrenamiento del modelo YOLOv8n se aplicó el nuevo conjunto de datos de la misma manera que en la **Tarea 5**.

Se aplicaron las prácticas MLOps para el guardado del control de versiones y el monitoreo de métricas de precisión y tasa de error en la detección de objetos, en código, datos y modelo mejorado (En Google Drive y Google Colaboratory).

### **5.2.2. Objetivo 2: Utilizar el modelo YOLOv8n con el dataset creado para medir la precisión y la tasa de error en la detección de sacos de balanceado de camarón Aquaxcel GROWER 5 (25 kg), Aquaxcel Pre-Starter 2 (25 kg), y ECOFEED Purina (25 kg).**

Este objetivo toma como referencia la fase de modelado y evaluación de CRISP-DM. La división de los datos incluye los subconjuntos de entrenamiento, validación y prueba, los primeros dos subconjuntos se utilizaron para el entrenamiento del modelo, mientras que el subconjunto de prueba se empleó para medir el rendimiento y obtener las métricas del modelo entrenado con el conjunto de datos creado.

Se utilizó Google Colab para cargar los datos y realizar la prueba de rendimiento, se creó un script para evaluar los datos de prueba utilizando la librería Ultralytics y la función `val()` para obtener la métrica de precisión y la matriz de confusión, también proporciona métricas como recall, mAP50 y mAP50-95, y gráficos como precisión-confianza, precisión-recuperación, y recuperación-confianza.

El modelo YOLOv8 para sacar las métricas de mAP50 y mAP50-95<sup>13</sup> debe realizar la ejecución del modelo con los datos de pruebas, para obtener las predicciones de los objetos detectados en cajas delimitadoras, con ello se utiliza la forma a continuación:

$$IoU = \frac{\text{Área de Intersección}}{\text{Área de Unión}} \quad (1)$$

---

<sup>13</sup> [YOLO Métricas de rendimiento - Ultralytics YOLO Docs](#)

Las predicciones que salen como resultado del modelo se ordenan de acuerdo con su puntaje de confianza, y con ellas se utiliza la precisión y el recall como la proporción de verdaderos positivos sobre el total de verdaderos positivos más los falsos negativos, en el cual se obtiene la curva de la Precisión-Recall, en donde se calcula el área bajo esta curva, conocida como Average Precision (AP). Con ello se realiza el enfoque con las métricas de mAP50 y mAP50-95, que básicamente utilizan el IoU de acuerdo con el umbral enfocado, si es mAP50 el IoU tiene que ser menos de 0.5 y, en cambio, el mAP50-95 el IoU se calcula con los valores de AP para varios umbrales de IoU, desde 0.5 hasta 0.95 en incrementos de 0.05, y luego se promedian estos valores, dando como respuesta a esta métrica.

La tasa de error se calculó manualmente porque la función val() de Ultralytics no proporciona este dato, se extrajeron los valores de Verdaderos Positivos (TP), Verdaderos Negativos (TN), Falsos Positivos (FP) y Falsos Negativos (FN) de la matriz de confusión no normalizada, con esto se calculó la tasa de error utilizando la siguiente fórmula<sup>14</sup> para cada clase individual como para el modelo en general:

$$Tasa\ de\ Error = \frac{(FP + FN)}{(TP + TN + FP + FN)} \quad (2)$$

Todo el proceso fue documentado conforme a las prácticas MLOps para asegurar la reproducibilidad y el control de versiones.

### 5.3. Recursos

Los materiales utilizados a lo largo del proyecto de Trabajo de Integración Curricular, se desglosan con los siguientes recursos:

#### 5.3.1. Recursos Metodológicos

##### 5.3.1.1. Metodología Experimental

- **CRISP-DM**

Proporcionó una estructura clara para el proyecto asegurando que las fases de comprensión de los datos, preparación de los datos, y modelado del proceso fueron debidamente ejecutadas y documentadas dentro del documento TIC.

---

<sup>14</sup> [Evaluación de rendimiento de un modelo de machine learning - DEV Community](#)

- **Prácticas MLOps**

Se realizó el seguimiento de cambios en los datos y el código mediante Github y Google Drive, monitoreo de métricas para evaluar el rendimiento del modelo y la automatización de procesos a través de scripts para asegurarse del control de versiones y la calidad del modelo

### **5.3.1.2. Ajuste Fino (Fine Tuning)**

Fue utilizado para mejorar el rendimiento del modelo YOLOv8n en la detección de sacos de balanceado de camarón ajustando los hiperparámetros, aplicando nuevos datos al entrenamiento y la evaluando iterativamente del modelo, ajustando según el rendimiento medido en los conjuntos de datos de validación y prueba.

### **5.3.2. Recursos técnicos**

Se realizaron dos entrevistas, para entrar en contexto para el proyecto TIC y el problema a resolver:

#### **5.3.2.1. Entrevista a la empleada de la empresa ASOCAM.**

La Ing. Marjorie Aponte Suárez, encargada de la contabilidad y administración en ASOCAM, describió los procesos logísticos de la empresa, dando un enfoque más específico a los productos dentro de la empresa junto con su despacho, por sus problemas tanto en la cuenta de los productos del estado anterior y después de un despacho, los productos con características similares, y la verificación exitosa en el despacho a los clientes. Marjorie afirmó que el principal problema es la continua adopción de la administración manual, causando confusiones con la gestión y despacho de muchos productos. Por la inexistencia de adquirir tecnologías especializadas dentro de las instalaciones que facilitarían la gestión del mismo (Ver **Anexo 2**).

#### **5.3.2.2. Entrevista al experto en Datasets.**

El Ing. Miguel Cumbicus Pineda Mg. Sc., docente en la Carrera de Computación de la Universidad Nacional de Loja, menciona la importancia de los datasets en el mundo del machine learning y por supuesto en la adopción y fundamental en la detección de objetos mediante la visión por computadora, afirmando que sin datos no se pueden desarrollar modelos de inteligencia artificial. Oscar Cumbicus describió el proceso de recolección, condiciones necesarias para la captura, y el procesado de los datos posterior a la

recolección. Además, recomendó herramientas como OpenCV, Roboflow y Labellmg para la anotación de datasets (Ver **Anexo 1**).

### **5.3.3. Recursos Tecnológicos**

#### **5.3.3.1. Hardware**

- **Computadora Portátil**

Se utilizó la computadora portátil personal del autor del documento, para la gestión de datos, desarrollo técnico y realización del documento TIC y cumplimiento académico de la Universidad Nacional de Loja.

- **Teléfono móvil.**

El teléfono móvil se utilizó solo la cámara para poder realizar la captura de datos, mediante el modelo 2201117TG con una apertura de f/1.8 y el tamaño en píxel de las imágenes de 50 megapíxeles.

#### **5.3.3.2. Software**

- **Python**

Lenguaje de programación utilizado, por su facilidad de uso y librerías que permiten una escritura de código más simple, además, que es necesario y recomendado para el procesado de los datos, y el entrenamiento de modelos de machine learning y deep learning. Se utilizó este lenguaje para el procesamiento de datos, entrenamiento y prueba del modelo con el conjunto de datos.

- **Visual Studio Code**

El IDE Visual Studio Code, sirvió para la escritura de código de una manera que facilita el entendimiento y estructura de forma local, específicamente utilizado para el redimensionamiento de imágenes y la modificación del archivo yaml del entrenamiento del modelo YOLOv8n.

- **Google Colaboratory**

Esta herramienta proporciona recursos de hardware necesarios para el desarrollo y entrenamientos modelos, como es en este caso YOLOv8n, por la gran cantidad de datos que se posee. Es una herramienta recomendada por el Ing. Oscar Cumbicus para el procesado de datos y entrenamiento de modelos de IA.

- **Labellmg**

Aplicación de escritorio utilizada para la anotación supervisada de imágenes del conjunto de datos especializado creado. Con la creación de cuadros delimitadores alrededor de los objetos de interés (sacos de balanceado de camarón) y guardados con el formato de anotación YOLO que es un archivo txt.

- **Google Drive**

Recurso indispensable para el manejo de versiones del conjunto de datos, que utiliza la nube que sirve como repositorio para los usuarios de Google.

- **Github**

Github es una plataforma web, que permitió compartir código realizado en todas las fases, facilitando el control y gestión de versiones del mismo.

#### **5.4. Participantes**

Las personas involucradas en el presente proyecto TIC, del itinerario de sistemas inteligentes fueron las siguientes:

- El estudiante autor del presente documento del Trabajo de Integración Curricular (TIC) Keyner Alexis Manchay Montoya, cuál inició las tareas desde el planteamiento de la propuesta Proyecto Integrador Curricular (PIC), hasta el desarrollo y culminación de los objetivos del presente trabajo.
- El director responsable del presente TIC es el Ing. Oscar Miguel Cumbicus Pineda Mg. Sc., el cual ayudó al autor a realizar las revisiones del documento y la parte técnica, para el cumplimiento y desarrollo del proyecto presente.
- La administradora del Departamento de Contable en la Asociación de Productores Camaroneros (ASOCAM), Ing. Marjorie Aponte Suarez, por facilitar la información necesaria para el TIC.

## 6. Resultados

En esta sección se detallan los resultados obtenidos para cada uno de los objetivos específicos, junto con las tareas realizadas para cumplir con dichos objetivos. En la **Sección 6.1.** se explica la construcción y preparación del conjunto de datos de imágenes, para su entrenamiento y mejora del modelo YOLOv8n con la técnica de fine tuning para la detección de objetos. La **Sección 6.2.** se evalúa el modelo entrenado con el subconjunto de datos de prueba, tomando las métricas de precisión y tasa de error.

**6.1. Objetivo 1: Recolectar imágenes de sacos de balanceado de camarón Aquaxcel GROWER 5 (25 kg), Aquaxcel Pre-Starter 2 (25 kg) y ECOFEED Purina (25 kg) para anotarlas y formar un dataset diverso y representativo, tomando como referencia la metodología CRISP-DM y las prácticas de MLOps.**

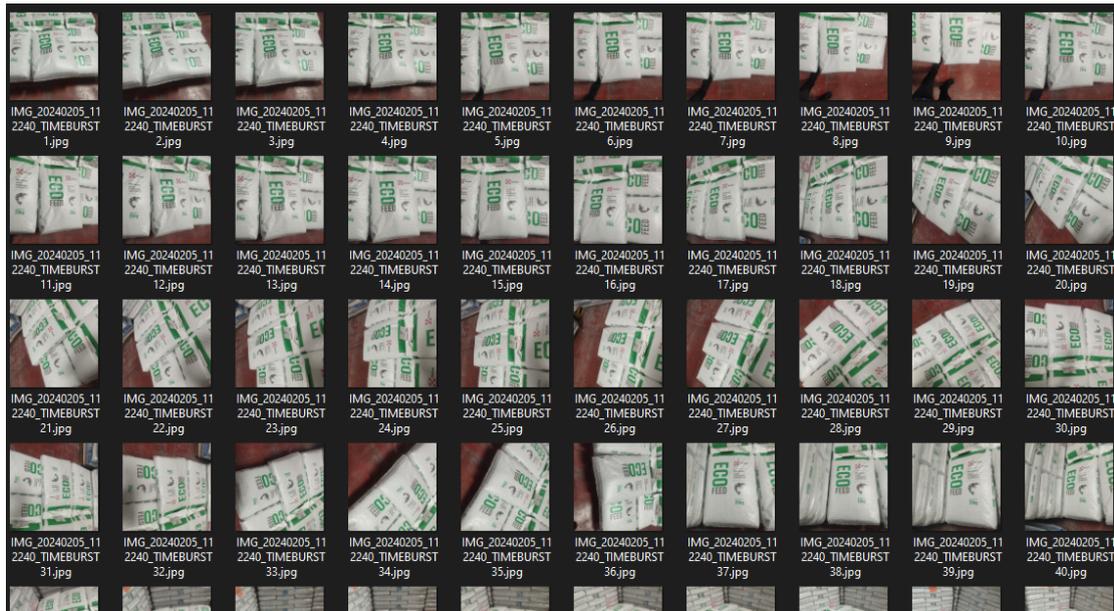
**6.1.1. Tarea 1:** Captura de datos mediante fotografías a los sacos de balanceado de camarón en diversas condiciones y entornos.

La recolección de datos se llevó a cabo considerando los escenarios disponibles dentro de las instalaciones de la empresa (ASOCAM) aprovechando la iluminación, y contrastes con otros sacos para añadirle una mayor diferencia en sus escenarios, utilizando los recursos expuestos en la **Tabla 2** como la cámara de celular y sacos de balanceado de camarón.

Llevando a cabo el proceso de captura se tomaron en consideración el contenido de la **Tabla 3** donde se especifican las condiciones necesarias para obtener datos brutos con la mejor calidad posible<sup>15</sup>, por lo que se obtuvo como resultado 3280 datos (imágenes) con dimensión de 3072 x 3072, con resolución horizontal y vertical de 72 ppp (puntos por pulgada), un rango de peso de imagen de 1.52 MB a 4.58 MB, como se muestra en la **Figura 8** donde se puede observar el conjunto de datos recolectado.

---

<sup>15</sup> [Conjunto de datos V1 - Datos Recolectados - Google Drive](#)



**Figura 8.** Conjunto de datos de imágenes recolectados de Sacos de balanceado de camarón.

### 6.1.2. Tarea 2: Preparación de los datos mediante la clasificación, criterios de inclusión y exclusión, limpieza y redimensionamiento.

Con los datos recolectados, en esta tarea se procedió a procesar los datos recolectados, con el objetivo de tener un conjunto de datos limpios y listos para la siguiente fase de anotación de los datos (**Sección 6.1.3.**). Esta tarea es importante, porque se define la calidad del resultado final, ya sea de un conjunto de datos o un modelo de deep learning, como se especifica en **TR09** que menciona lo más importante para la IA son los datos, para que su uso en los modelos especializados no se vean afectados.

#### **Clasificación y Separación de Datos:**

Se inició por separar los datos en 3 subconjuntos diferentes de clases, por lo que se optó por utilizar el nombre original del saco para esta tarea, como: Aquaxcel GROWER 5 (25 kg), Aquaxcel Pre-Starter 2 (25 kg) y ECOFEED Purina (25 kg). Teniendo en cuenta que en algunas imágenes se encuentran más de una clase, se toma la decisión de separar los datos, por las imágenes que tengan una clase más resaltante, ósea que tengan mayor cantidad de sacos en una imagen o que abarque un considerable espacio dentro de la misma, por lo que se eliminan las imágenes que no se encuentren en ninguna de estas, como es el caso de la **Figura 9.**



**Figura 9.** Imagen descartada por no pertenecer a ninguna clase.

En total se eliminaron 4 imágenes del conjunto de datos que no pertenecen ninguna de las clases que se definieron en la anterior clasificación<sup>16</sup> (ver **Tabla 6**).

**Tabla 6.** Cantidad de datos totales según su clase, después de la eliminación de datos.

Nombre de los datos	Cantidad de Datos
Cantidad de datos totales	3276
Datos de la Clase 1 (Aquaxcel GROWER 5 (25 kg))	988
Datos de la Clase 2 (Aquaxcel Pre-Starter 2 (25 kg))	1107
Datos de la Clase 3 (ECOFEED Purina (25 kg))	1181
Datos que no se encuentran en ninguna clase (Descartadas)	4

### **Criterios de Inclusión y Exclusión:**

Posteriormente, se establecieron los criterios de inclusión y exclusión que se pueden observar en la **Tabla 4**, con el objetivo de determinar qué imagen de todo el conjunto de datos está en condiciones suficientes de calidad para que puedan pasar a la siguiente tarea,

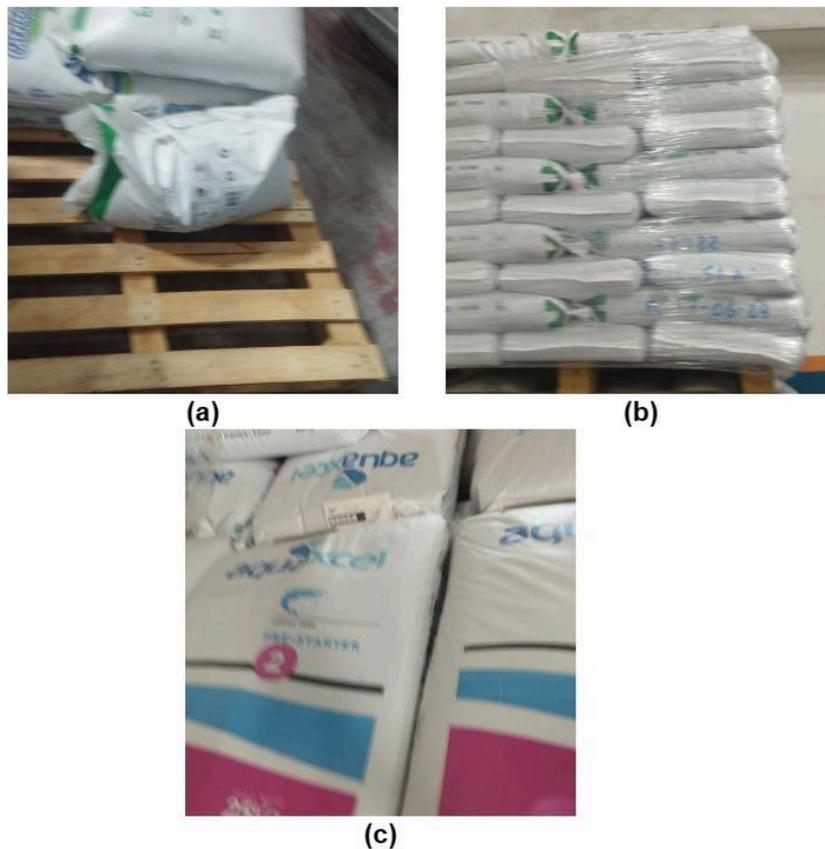
<sup>16</sup> [Conjunto de datos V2 - Clasificación y separación de Datos - Google Drive](#)

este proceso es importante, porque se descartan imágenes que no aportan peso al conjunto de datos, en criterios más estrictos las imágenes que no satisfagan estos criterios pueden malograr todo conjunto de datos en general.

### **Limpieza de Datos:**

Se comienza con la limpieza de datos con la clasificación manual del autor del presente documento TIC, al conjunto de datos, para comprobar que cada imagen individual cumpla con cada criterio definido en la **Tabla 4**.

En la **Figura 10** se muestran algunos ejemplos de imágenes que no cumplen con los criterios de inclusión y exclusión, habiendo un total de 131 imágenes que son parte de esta clasificación negativa, lo que se procedió a su eliminación.

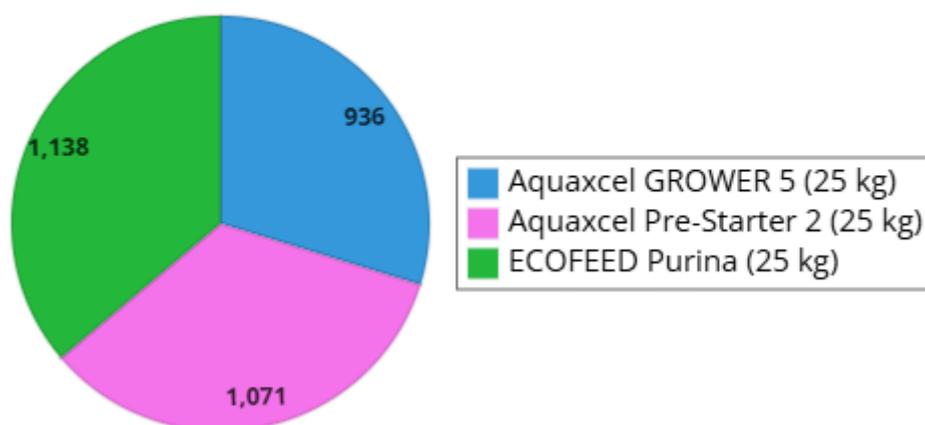


**Figura 10.** Imágenes que no cumplen con los criterios de inclusión y exclusión. (a) Imagen muy borrosa que no se distinguen las características del saco. (b) Imagen borrosa y donde el forro de plástico no ayuda a poder visualizar las características del saco. (c) Imagen donde el cuerpo del saco objetivo no puede verse al menos un 75%.

Una vez realizada la limpieza de los datos con la ayuda de los criterios de inclusión y exclusión, el conjunto de datos en esta versión contiene, 3145 elementos entre las 3 clases<sup>17</sup>, como se muestra en la **Tabla 7** y la **Figura 11**.

**Tabla 7.** Cantidad total del conjunto de datos, después de la limpieza.

Nombre de los datos	Cantidad de Datos
Cantidad de datos totales	3145
Datos de la Clase 1 (Aquaxcel GROWER 5 (25 kg))	936
Datos de la Clase 2 (Aquaxcel Pre-Starter 2 (25 kg))	1071
Datos de la Clase 3 (ECOFEED Purina (25 kg))	1138



**Figura 11.** Gráfico que muestra la cantidad de las Imágenes, según su clase.

### Redimensionamiento de Imágenes:

Con el conjunto de datos limpio se procedió a revisar los requerimientos de entrada del modelo elegido (YOLOv8n), el cual se especializa en la detección de objetos y su configuración se muestra en la **Tabla 5**, lo que indica un tamaño de imagen (size) de 640 x 640 píxeles tanto para entrenamiento como para validación; mAP (mean Average Precision) es una métrica que mide la precisión promedio del modelo a lo largo de diferentes umbrales de IoU (Intersect over Union), el cual muestra un valor de 37.3 que indica la precisión media del modelo en un rango de IoU de 0.50 a 0.95; el siguiente apartado que es Speed CPU

<sup>17</sup> [Conjunto de datos V3 - Limpieza a los datos - Google Drive](#)

ONNX (Open Neural Network Exchange) medido en milisegundos (ms) muestra el tiempo promedio que toma al modelo realizar una predicción en una CPU, utilizando el formato ONNX, que se refiere que tarda aproximadamente 80.4 milisegundos por imagen, y en comparación con los demás modelos de la misma versión de YOLOv8 que son más robustos este es el más rápido; Speed A100 TensorRT (ms) este apartado indica el tiempo que toma realizar una predicción en una GPU Nvidia A100, utilizando la tecnología TensorRT de Nvidia para optimización, lo que significa que es significativamente menor en la GPU (0.99 milisegundos), mostrando la ventaja del uso de hardware especializado para tareas de procesamiento intensivo; Params (M) se refiere al número total de parámetros entrenables en el modelo, expresado en millones, lo que quiere decir que un mayor número de parámetros puede indicar un modelo más complejo y potencialmente más capaz, pero también más demandante en términos de recursos computacionales, YOLOv8n por ser el modelo más ligero es el que menos parámetros soporta; y finalmente FLOPs (Floating Point Operations Per Second) que indica la cantidad de operaciones de punto flotante que el modelo necesita realizar durante un paso de inferencia, el cual para este modelo en específico muestra un valor de 8.7 billones, que ayuda a entender la complejidad computacional del modelo.

Conociendo los parámetros que utiliza el modelo YOLOv8n, se realizó el redimensionamiento de los datos de forma local, utilizando el IDE Visual Studio code, en lenguaje de programación Python, en la **Figura 12**, se muestra la porción de código que establece el ancho y la altura de la imagen que corresponde a 640 x 640 píxeles.

```
width = 640
height = 640
dim = (width, height)
```

**Figura 12.** Porción de código que establece las dimensiones de la imagen según el modelo YOLOv8n.

La continuación del código anterior se plasma en la **Figura 13**, que muestra un ciclo repetitivo por el “for” que funciona tomando las rutas de las imágenes que van entrando, para que se redimensionen con los parámetros mostrados en la **Figura 12**, para luego ser guardados en la ruta de salida.

```

for image_path in image_paths:
    # Cargar la imagen
    image = cv2.imread(image_path)

    # Redimensionar la imagen
    resized_image = cv2.resize(image, dim, interpolation=cv2.INTER_AREA)

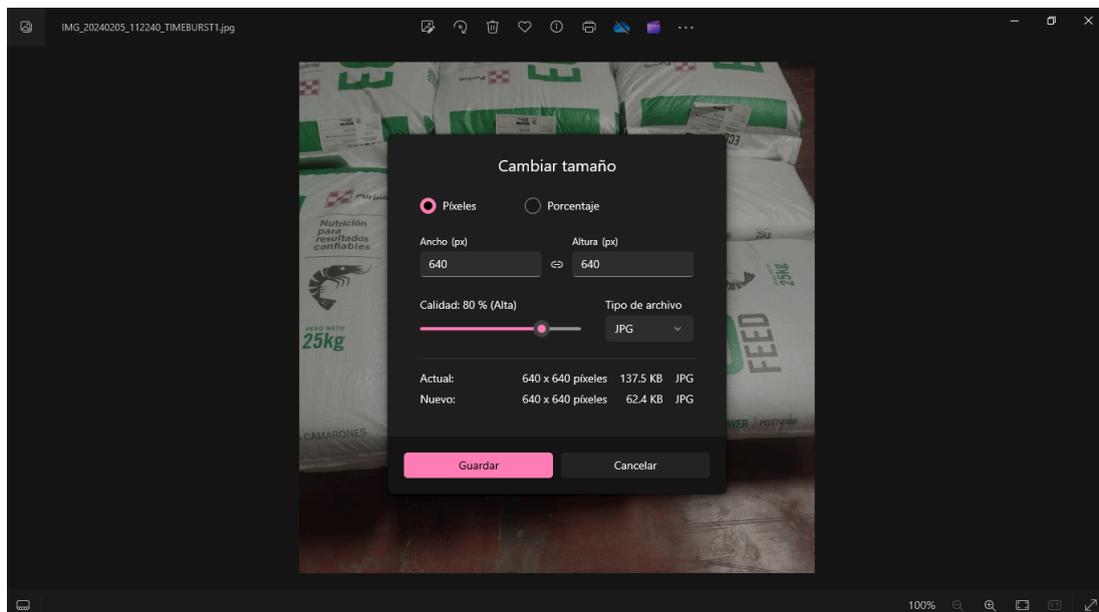
    # Construir el path de salida
    output_path = os.path.join(output_directory, os.path.basename(image_path))

    # Guardar la imagen redimensionada
    cv2.imwrite(output_path, resized_image)

```

**Figura 13.** Porción de código que redimensiona las imágenes del conjunto de datos.

El resultado del redimensionamiento<sup>18</sup> del código<sup>19</sup> mostrado de las **Figuras 12** y **Figura 13** se muestran a continuación en la **Figura 14** una imagen que viene de parte del conjunto de datos en el formato JPG con dimensiones de 640 x 640 píxeles.



**Figura 14.** Presentación en un visualizador de imágenes del resultado del redimensionamiento del conjunto de datos.

Con la configuración del tamaño de las imágenes realizada, y con un equilibrio entre la cantidad de las imágenes (clases) aceptable, como se muestra en la **Tabla 7**, no es necesario la implementación de alguna técnica de aumento de datos, ya que el conjunto de datos es adecuado.

<sup>18</sup> [Conjuntos de datos V4 - Redimensionamiento de los datos - Google Drive](#)

<sup>19</sup> [Código de Redimensionamiento de Imágenes - GitHub](#)

## Renombramiento de Imágenes:

Para tener una organización en cuanto a los datos se refiere, es necesario asignarle un renombramiento a las imágenes, para que cada dato tenga un nombre que sea identificativo y esté organizado, por lo que se optó por utilizar el formato "Imagen\_x", donde "x" representa un número ascendente que comienza desde 1, por lo que se utilizó Google Colab para esta tarea, por la cualidad de sus recursos computacionales que pueden ser necesarios para el procesamiento; entonces se subieron las clases por separados por carpetas (ver **Figura 15**), además que se implemente un contador que permite la continuación numérica en los nombres.

```
# Ruta de las carpetas con imágenes
folders = ['/content/Aquaxcel GROWER 5 (25 kg)', '/content/Aquaxcel Pre-Starter 2 (25 kg)', '/content/ECOFEED Purina (25 kg)']

# Inicializar contador
counter = 1
```

**Figura 15.** Presentación de la declaración de las variables como carpetas de las clases y el contador.

Posteriormente, se crea un script para utilizar ciclos anidados para renombrar las imágenes de todas las clases, por lo que este proceso comienza desde la clase "Aquaxcel GROWER 5 (25 kg)", seguido por "Aquaxcel Pre-Starter 2 (25 kg)" y "ECOFEED Purina (25 kg)" realizando la numeración desde ese orden, en la **Figura 16** se muestra esta porción de código.

```
for folder in folders:
    # Obtener lista de archivos en la carpeta
    image_files = sorted(os.listdir(folder))

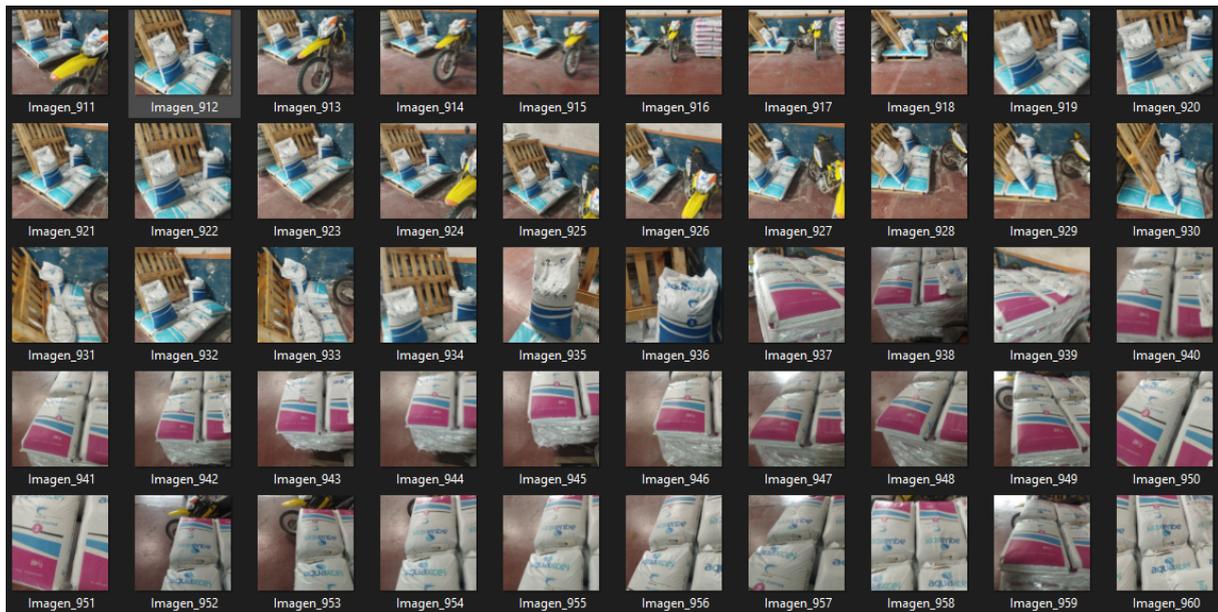
    # Filtrar solo archivos .jpg
    image_files = [f for f in image_files if f.lower().endswith('.jpg')]

    # Renombrar archivos
    for filename in image_files:
        # Crear nuevo nombre de archivo
        new_filename = f"Imagen_{counter}.jpg"
        # Obtener ruta completa de archivo actual y nuevo archivo
        src = os.path.join(folder, filename)
        dst = os.path.join(folder, new_filename)
        # Renombrar archivo
        os.rename(src, dst)
        # Incrementar contador
        counter += 1

print("Renombrado completado.")
```

**Figura 16.** Presentación del script utilizado para el renombramiento de los datos.

Con ello se obtiene los resultados desde las rutas de las carpetas expuestas anteriormente<sup>20</sup>, con el formato de “Imagen\_x”, que tomando como ejemplo la última imagen de la primera clase (Aquaxcel GROWER 5 (25 kg)), se renombra a “Imagen\_936”, para que luego continúe la numeración gracias al contador desde la siguiente carpeta, comenzando su primera imagen con el nombre de “Imagen\_937”, en la **Figura 17** se muestra este resultado<sup>21</sup>.



**Figura 17.** Presentación desde la carpeta del conjunto de datos renombrados.

### 6.1.3. Tarea 3: Anotación de los datos con la herramienta Labellmg.

Con los datos procesados y organizados, se realiza la fase de la anotación de las imágenes en cuadros delimitadores, que actualmente se tiene 3145 elementos en total, como se muestra en la **Tabla 7**, además del renombramiento que ayudo a tener un control más preciso en la asignación de las anotaciones.

La anotación se realizó de forma manual por criterio visual del autor del presente documento, tomando como referencia los criterios de inclusión y exclusión de la **Tabla 4** aplicados a cada objeto dentro de cada imagen anotada, este proceso se llevó a cabo con la herramienta Labellmg, el cual es un software de escritorio de código abierto, que es ampliamente utilizado para la anotación manual de imágenes de detección de objetos, permitiendo un reconocimiento eficiente de los objetos de interés por su interfaz que facilita la creación de cuadros delimitadores alrededor de cada saco de balanceado, sin mencionar que es capaz soportar múltiples formatos de archivo de anotación, lo que ayuda a la

<sup>20</sup> [Código Renombramiento de datos - Colab](#)

<sup>21</sup> [Conjuntos de datos V5 - Renombramiento de los datos - Google Drive](#)

compatibilidad en los datos para diversos modelos de detección de objetos, y para este proyecto en particular se utilizó el formato de YOLO<sup>22</sup>, que se compone dentro de un archivo “txt” como se puede evidenciar en la **Figura 18**, las clases, por lo tanto, se las definieron según se clasifica en la **Sección 6.1.2.** como es “Aquaxcel GROWER 5 (25 kg)”, “Aquaxcel Pre-Starter 2 (25 kg)”, y “ECOFEED Purina (25 kg)” (ver **Tabla 8**).

```
0 0.217969 0.179688 0.398438 0.356250
0 0.459375 0.741406 0.515625 0.501563
0 0.114844 0.625000 0.226562 0.609375
0 0.584375 0.278906 0.393750 0.354687
0 0.753906 0.673438 0.492188 0.575000
2 0.867188 0.078906 0.265625 0.154688
2 0.582812 0.060937 0.287500 0.118750
```

**Figura 18.** Archivo de anotación “Imagen\_566” en formato YOLO (txt).

El primer número de la izquierda de la **Figura 18** es el índice que compone una clase, por lo que esta imagen tendría 5 de la clase Aquaxcel GROWER 5 (25 kg) y 2 de la clase ECOFEED Purina (25 kg) según su etiqueta.

**Tabla 8.** Índices generados al crear las clases dentro de la herramienta LabelImg.

Clase	Índice
Aquaxcel GROWER 5 (25 kg)	0
Aquaxcel Pre-Starter 2 (25 kg)	1
ECOFEED Purina (25 kg)	2

En la **Figura 19** se muestra como se aplicaron los cuadros delimitadores a cada una de las 3 clases que se definieron en la **Tabla 8**, y como se puede observar se encuentran en entornos parecidos, pero diferentes, lo que ayuda a poder clasificar de mejor manera por sus características y no por sus entornos levemente parecidos, que en la mayoría de las imágenes se tiene una sola clase resaltante de saco de balanceado de camarón definida, pero en algunas de ellas se tiene una mayor cantidad de clases, pero que claramente no destacan con respecto a la clase predominante, que puede ser la que tiene una mayor ocupación de espacio en la imagen o tiene una mayor cantidad de su clase, en el cual se debe etiquetar con precisión para no confundir las anotaciones con características ajenas no correspondiente a cada clase, para que los modelos de deep learning implementados más adelante no tengan pesos erróneos que malogren la calidad del modelo.

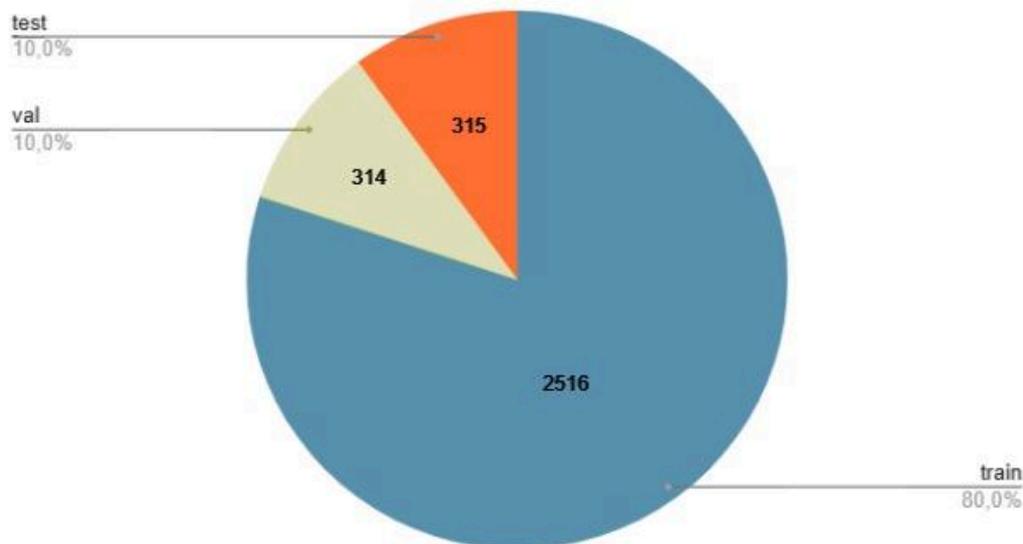
<sup>22</sup> [Conjuntos de datos V6 - Anotación de los datos - Google Drive](#)



**Figura 19.** Imágenes con objetos dentro de cuadros delimitadores. (a) Se etiquetaron dos objetos de la clase 1. (b) Se etiquetaron tres objetos de la clase 2. (c) Se etiquetaron cinco objetos de la clase 3.

#### 6.1.4. Tarea 4: División del conjunto de datos entre entrenamiento (train), validación (val) y prueba (test).

En la división de datos, se establecieron las proporciones de entre un 80% en el split de “train” (entrenamiento), 10% en el split de “val” (validación) y para el split de “test” (prueba) un 10%, como se muestra en la **Figura 20** donde se pueden observar las proporciones de los datos según su subsección.



**Figura 20.** División del conjunto de datos.

Para este cometido se utilizó la herramienta Google Colab<sup>23</sup> y se implementó código en Python. La **Figura 21** muestra cómo se realiza la división de los datos en tres rutas diferentes, cada una con una carpeta de “images” para las imágenes y “labels” para las anotaciones. Los subconjuntos resultantes son: entrenamiento (80%), validación (10%) y prueba (10%). En el código primero, se calcula el número total de imágenes y se determinan los índices correspondientes para cada subconjunto. Luego, las imágenes se distribuyen aleatoriamente en las respectivas carpetas, también se define una función que copia las imágenes y sus etiquetas correspondientes a las carpetas de destino, esta función toma como parámetros la lista de imágenes y la carpeta de destino, y copia cada imagen junto con su archivo de etiqueta. Finalmente, se llama a esta función para copiar los archivos a las carpetas de entrenamiento, validación y prueba.

```

# Dividir los datos en proporciones 80% train, 10% val, 10% test
total = len(imagenes)
train_split = int(total * 0.8)
val_split = int(total * 0.9) # 80% train + 10% val = 90%

train_imgs = imagenes[:train_split]
val_imgs = imagenes[train_split:val_split]
test_imgs = imagenes[val_split:]

# Función para copiar archivos a la carpeta correspondiente
def copiar_archivos(lista_imgs, carpeta_destino):
    for img in lista_imgs:
        base_name = img.replace('.jpg', '')
        shutil.copy(os.path.join(ruta_imagenes, img), os.path.join(carpeta_destino, 'images', img))
        shutil.copy(os.path.join(ruta_etiquetas, base_name + '.txt'), os.path.join(carpeta_destino, 'labels', base_name + '.txt'))

# Copiar los archivos a las carpetas de train, val y test
copiar_archivos(train_imgs, ruta_train)
copiar_archivos(val_imgs, ruta_val)
copiar_archivos(test_imgs, ruta_test)

```

**Figura 21.** Script para la división de datos en 3 subconjuntos: train (80%), val (10%) y test (10%).

<sup>23</sup> [Código División de datos - Colab](#)

Como resultado se tiene los 2516 datos de entrenamiento, en 314 datos de validación, y en prueba 315 datos<sup>24</sup> (ver **Tabla 9**).

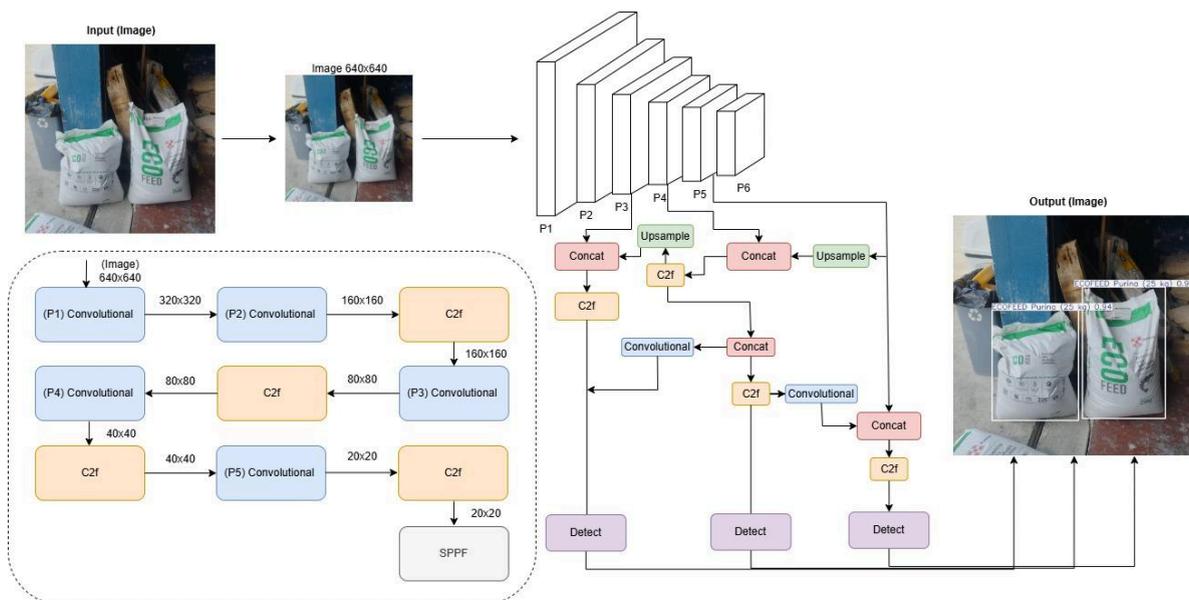
**Tabla 9.** División de los datos según su subconjunto.

Nombre del subconjunto de datos	Cantidad de datos
Datos totales	3145
Datos de entrenamiento (train)	2516
Datos de validación (val)	314
Datos de prueba (test)	315

### 6.1.5. Tarea 5: Entrenamiento del modelo YOLOv8n.

El entrenamiento del modelo YOLOv8n, para la detección de objetos con el conjunto de datos preparado anteriormente, se inició con la definición de los hiperparámetros y la creación del archivo yaml, para posteriormente realizar su ejecución para obtener su matriz de confusión y revisar su rendimiento.

En la **Figura 22** se puede observar la representación de la arquitectura del modelo YOLOv8 y como este toma las características de los datos para realizar las predicciones en la detección de los sacos de balanceado de camarón, procedimiento que se llevara a cabo con la ejecución del entrenamiento.



**Figura 22.** Diagrama del modelo YOLOv8 para la detección de sacos de balanceado de camarón.

<sup>24</sup> [Conjuntos de datos V7 - División de datos - Google Drive](#)

## Configuración de Hiperparámetros:

Se llevó a cabo utilizando el framework YOLOv8 en Google Colab, aprovechando los recursos de cómputo en la nube para manejar grandes volúmenes de datos y realizar el entrenamiento necesario. Se manejó el dataset previamente preparado y dividido en conjuntos de entrenamiento, validación y prueba, de las anteriores tareas; el proceso de entrenamiento incluye la configuración de los parámetros del modelo, que se compone por la task (tarea) que se aplicó “detect” que significa que es para la detección de objetos, mode (modo) que se trata del “train” como su nombre indica el entrenamiento del conjunto de datos, el número de epochs (épocas) el cual se aplicaron 50 por el tamaño no muy grande del conjunto de datos, data (datos) es la ruta del archivo yaml “/content/data1/data.yaml” que contiene las rutas de los subconjuntos de datos de entrenamiento y validación sin mencionar la declaración de las clases, model (modelo) el modelo a utilizar para en el entrenamiento que es “yolov8n.pt”, imgsz (tamaño de imagen) el cual se recomienda 640 y es el aplicado al conjunto de datos en la **Tarea 2**, y el batch (tamaño de lote) se aplicó 16 por la utilización de Google Colab (recursos computacionales) y el tamaño del conjunto de datos (mediano por tener miles de datos) que permite esta asignación, en la **Tabla 10** se pueden observar de una manera más sencilla los hiperparámetros asignados para el entrenamiento del modelo.

**Tabla 10.** Configuración de los hiperparámetros para el entrenamiento de los datos con el modelo *yolov8n.pt*.

Nombre del hiperparámetro	Valor del Hiperparámetro
task	detect
mode	train
data	/content/data1/data.yaml
model	yolov8n.pt
imgsz	640
batch	16

## Ejecución del Ciclo de Entrenamiento:

A continuación, se cargaron los datos a Google Colab<sup>25</sup>, y se dio inicio a la ejecución del ciclo de entrenamiento. Durante cada época, el modelo se entrenó utilizando el conjunto

<sup>25</sup> [Código Entrenamiento de datos con YOLOv8n - Colab](#)

de datos de entrenamiento y se validó utilizando el conjunto de datos de validación (ver **Figura 23**).

```

train: /content/data1/train
val: /content/data1/val
test: /content/data1/test
nc: 3
names:
  - Aquaxcel GROWER 5 (25 kg)
  - Aquaxcel Pre-Starter 2 (25 kg)
  - ECOFEED Purina (25 kg)

```

**Figura 23.** Archivo yml, donde se declaran las rutas de los subconjuntos de datos, número de clases y los nombres de las clases.

En la **Tabla 11** se muestra el proceso de entrenamiento del modelo YOLOv8n, donde se observa la información correspondiente a las últimas iteraciones de las 50 épocas del ciclo de entrenamiento<sup>26</sup>. Los datos incluyen el uso de memoria GPU, la pérdida de la caja delimitadora (box\_loss), la pérdida de clasificación (cls\_loss), la pérdida de la red de detección (dfl\_loss), el tamaño de los lotes procesados, y las métricas de precisión (mAP50 y mAP50-95). Alcanzando una precisión (mAP50) de 0.959, lo cual indica un alto nivel de precisión en la detección de los sacos de balanceado de camarón.

dssd

**Tabla 11.** Rendimiento del Modelo YOLOv8n durante el primer entrenamiento.

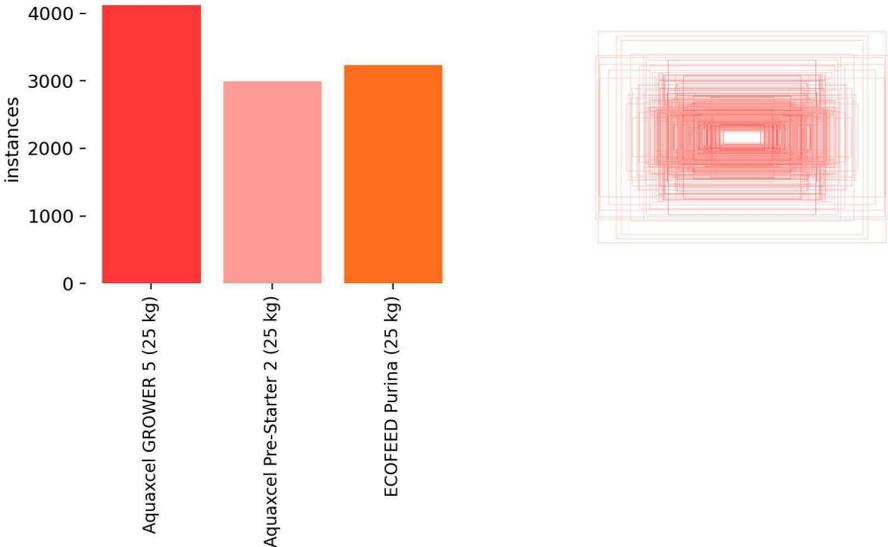
Epoch	box_loss	cls_loss	dfl_loss	mAP50	mAP50-95
1/50	1.178	2.196	1.339	0.564	0.349
...	...	...	...	...	...
47/50	0.6429	0.4315	0.9683	0.957	0.806
48/50	0.6367	0.4271	0.9631	0.959	0.81
49/50	0.633	0.4214	0.9599	0.957	0.809
50/50	0.6329	0.4226	0.9566	0.958	0.81

*Ciclo de entrenamiento (Épocas); Pérdida de Caja (box\_loss); Pérdida de Clasificación (cls\_loss); Pérdida de Regresión de Distancia (dfl\_loss); Precisión Media a IOU 0.50 (mAP50); Precisión Media a IOU 0.50-0.95 (mAP50-95).*

Revisando las métricas, se puede notar en la **Figura 24** que se muestra en la parte izquierda un gráfico de barras que indican el eje vertical, el número de instancias y en el eje

<sup>26</sup> [Modelo del EntrenamientoV1 - Google Drive](#)

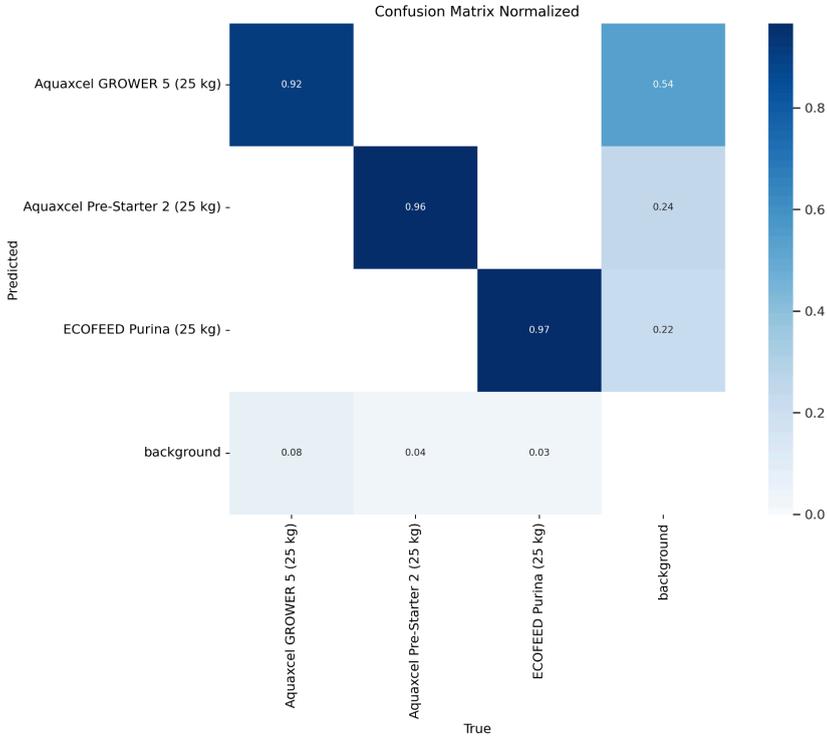
horizontal las clases de sacos de balanceado de camarón como lo son “Aquaxcel GROWER 5 (25 kg)”, “Aquaxcel Pre-Starter 2 (25 kg)”, y “ECOFEED Purina (25 kg)”. En cambio, en la parte derecha se presenta, la representación acumulativa de todas las cajas delimitadoras predichas por el modelo para las diferentes clases, lo que permite observar la concentración y dispersión de las detecciones, proporcionando una visión de cómo el modelo ha identificado y clasificado las diferentes instancias en el conjunto de datos. Teniendo esto en cuenta se puede notar en la misma figura una diferencia notable en la creación de los cuadros delimitadores, poseyendo la clase “Aquaxcel GROWER 5 (25 kg)” más de 4000 instancias, teniendo una diferencia aproximada de 1000 con las demás clases.



**Figura 24.** Cantidad de instancias en cada clase y la acumulación de las cajas delimitadoras del conjunto de datos después del primer entrenamiento del modelo.

Reforzando este desequilibrio en la matriz de confusión normalizada en la **Figura 25**, donde la primera clase tiene menor precisión y mayor pérdida en comparación de las otras clases. La matriz de confusión de las salidas de YOLOv8n muestra predicciones del modelo para las diferentes clases y el "background". La variable "background" se refiere a las áreas de las imágenes que no contienen ninguno de los sacos de balanceado de camarón que estamos tratando de identificar. Lo que se traduce en las partes de las imágenes que el modelo identifica correctamente como no pertenecientes a ninguna de las clases de interés; en la matriz de confusión normalizada, los valores en la fila horizontal "background" indican los falsos negativos (FN), que representan las instancias de las clases objetivo clasificadas incorrectamente como fondo. Por otro lado, los valores en la columna vertical "background" representan los falsos positivos (FP), que también son las instancias de fondo clasificadas incorrectamente como pertenecientes a una de las clases de sacos de balanceado de

camarón. Es decir que los valores en las celdas diagonales representan la proporción de instancias correctamente clasificadas para cada clase, mientras que los valores fuera de la diagonal indican errores de clasificación.



**Figura 25.** Matriz de confusión normalizada para la clasificación de sacos de balanceado de camarón utilizando el modelo YOLOv8n del primer entrenamiento.

**6.1.6. Tarea 6:** Mejora del conjunto de datos.

Identificando el problema de la **Tarea 5**, que trata de un desequilibrio notable en la cantidad de las instancias que afecta en el resultado de la precisión y perdida, como se evidencia en la **Figura 24** y **Figura 25**, se procedió en la acción de una mejora al conjunto de datos, tomando en consideración las tareas anteriores para una nueva limpieza en los datos, comenzando por la separación por clases del conjunto actual de datos (imagen y etiqueta).

La eliminación de datos se realizó según el criterio del autor, descartando imágenes y sus respectivas etiquetas de la clase con mayor número de instancias, “Aquaxcel GROWER 5 (25 kg)”. En total, se eliminaron 70 datos (ver **Tabla 12**), ya que esta clase tenía aproximadamente 1000 instancias adicionales<sup>27</sup>. Se eliminaron específicamente aquellas imágenes que contenían 10 o más cuadros delimitadores en una sola imagen.

<sup>27</sup> [Conjuntos de datos V8 - 70 datos eliminados - Google Drive](#)

**Tabla 12.** Datos actualizados del conjunto de datos después de la eliminación de imágenes de la mejora del conjunto de datos.

Nombre del subconjunto de datos	Cantidad de Datos
Datos totales	3075
Datos de la Clase 1 (Aquaxcel GROWER 5 (25 kg))	866
Datos de la Clase 2 (Aquaxcel Pre-Starter 2 (25 kg))	1071
Datos de la Clase 3 (ECOFEED Purina (25 kg))	1138

Posteriormente, se realizó el renombramiento de los datos, ya que la eliminación previa de datos había desconfigurado la numeración secuencial de las imágenes con sus etiquetas, generando saltos en los números. Se utilizó el código de la **Figura 16 (Tarea 2)**, con algunas modificaciones para incluir la carpeta “labels”. El código como se muestra en la **Figura 26** ahora renombra las imágenes y sus etiquetas correspondientes en pares, manteniendo el formato “Imagen\_x”, donde “x” es un número en orden ascendente<sup>28</sup>.

```
def rename_files(images_folder, labels_folder, prefix):
    images = sorted([f for f in os.listdir(images_folder) if f.endswith('.jpg')])
    labels = sorted([f for f in os.listdir(labels_folder) if f.endswith('.txt')])

    # Aseguramiento de que la cantidad de imágenes y etiquetas deben coincidir
    if len(images) != len(labels):
        print("Error: El número de imágenes y etiquetas no coinciden.")
        return

    for index, (image_name, label_name) in enumerate(zip(images, labels)):
        new_name = f"{prefix}_{index + 1}"

        # Renombramiento de imágenes
        image_src = os.path.join(images_folder, image_name)
        image_dst = os.path.join(images_folder, f"{new_name}.jpg")
        os.rename(image_src, image_dst)

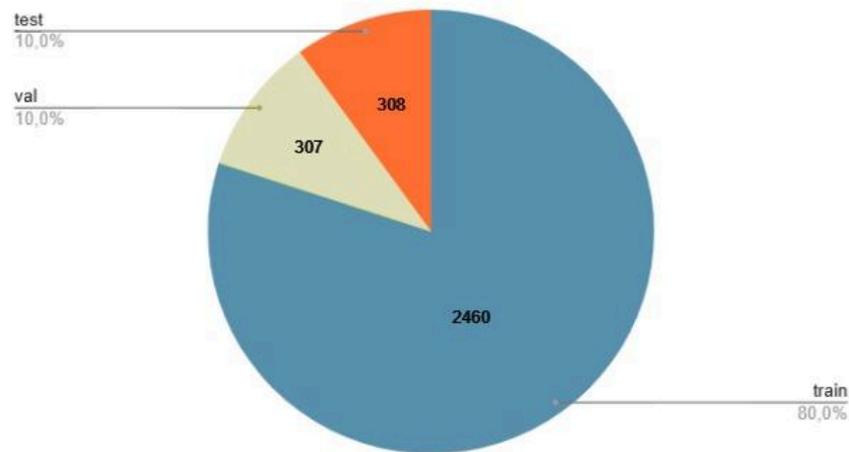
        # Renombramiento de etiquetas
        label_src = os.path.join(labels_folder, label_name)
        label_dst = os.path.join(labels_folder, f"{new_name}.txt")
        os.rename(label_src, label_dst)

    print(f'Renombramiento: {image_src} to {image_dst}')
    print(f'Renombramiento: {label_src} to {label_dst}')
```

**Figura 26.** Modificación del código de renombramiento de la tarea 2, para el renombramiento de imágenes junto con sus etiquetas.

<sup>28</sup> [Renombramiento Modificado - Colab](#)

Se reutilizó el código de división de datos de la **Tarea 4** para distribuir los datos en las carpetas de entrenamiento (80%), validación (10%) y prueba (10%)<sup>29</sup>, en la **Figura 27** se muestra la proporciones de los datos según su subconjunto.



**Figura 27.** División del conjunto de datos mejorado.

Luego, se implementó el mismo archivo “yami” (ver **Figura 22**) de la **Sección 6.1.5.** para que el modelo YOLOv8n detecte las diferentes clases y las rutas de los datos junto con sus etiquetas. Se realizó el entrenamiento con el nuevo conjunto de datos, con los mismos parámetros puesto en el anterior ciclo de la **Tabla 13**, con los siguientes resultados<sup>30</sup>:

**Tabla 13.** Rendimiento del Modelo YOLOv8n durante el segundo entrenamiento con el conjunto de datos mejorado.

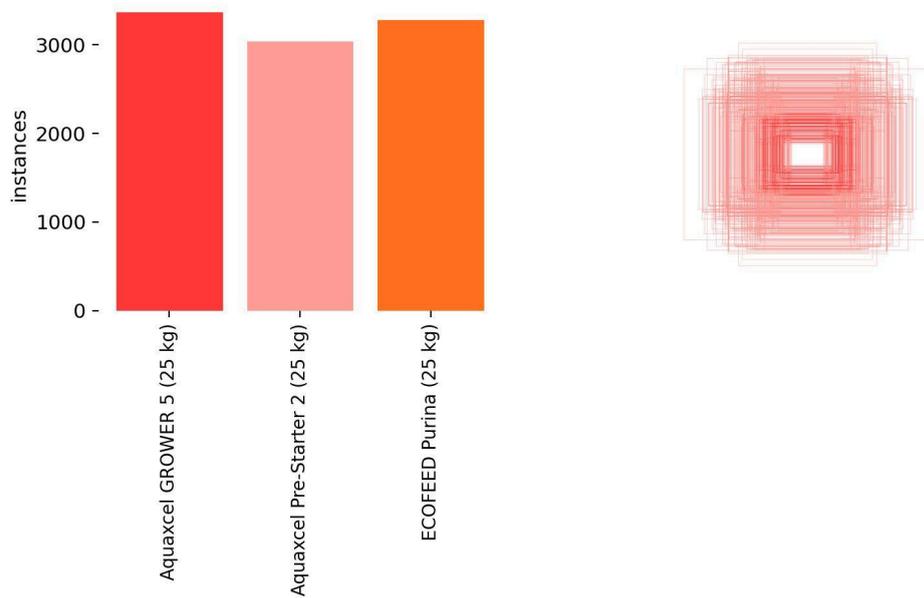
Epoch	box_loss	cls_loss	dfl_loss	mAP50	mAP50-95
1/50	1.18	2.190	1.319	0.533	0.37
...	...	...	...	...	...
47/50	0.6279	0.4294	0.9636	0.967	0.816
48/50	0.6275	0.4273	0.9613	0.965	0.82
49/50	0.6179	0.4179	0.9581	0.967	0.818
50/50	0.6149	0.4166	0.9535	0.967	0.82

*Ciclo de entrenamiento (Épocas); Pérdida de Caja (box\_loss); Pérdida de Clasificación (cls\_loss); Pérdida de Regresión de Distancia (dfl\_loss); Precisión Media a IOU 0.50 (mAP50); Precisión Media a IOU 0.50-0.95 (mAP50-95).*

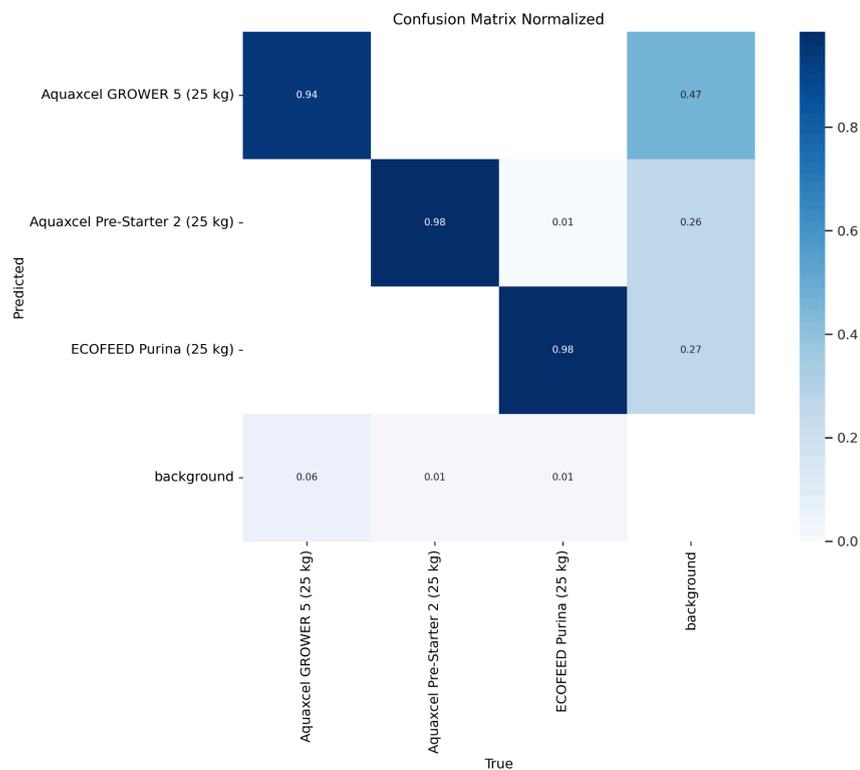
<sup>29</sup> [Conjuntos de datos V9 - División de datos - Google Drive](#)

<sup>30</sup> [Modelo Mejorado del EntrenamientoV2 - Google Drive](#)

Con los resultados de las métricas del entrenamiento, se muestra las instancias en la **Figura 28** mejor equilibradas y una matriz de confusión normalizada de mayor confianza por la precisión que muestra cada clase (ver **Figura 29**).



**Figura 28.** Cantidad de instancias en cada clase y la acumulación de las cajas delimitadoras del conjunto de datos mejorado después del segundo entrenamiento del modelo.



**Figura 29.** Matriz de confusión normalizada para la clasificación de sacos de balanceado de camarón utilizando el modelo YOLOv8n del segundo entrenamiento y el conjunto de datos mejorado.

**6.2. Objetivo 2: Utilizar el modelo YOLOv8n con el dataset creado para medir la precisión y la tasa de error en la detección de sacos de balanceado de camarón Aquaxcel GROWER 5 (25 kg), Aquaxcel Pre-Starter 2 (25 kg), y ECOFEED Purina (25 kg).**

Con el entrenamiento del modelo mejorado, se utilizó el subconjunto de datos de prueba (test) para medir el rendimiento del modelo y determinar la precisión y tasa de error obtenidos. Se optó por utilizar Google Colab, se empleó la función “val” de la librería Ultralytics, que se usa principalmente para obtener métricas del subconjunto de datos de validación, pero que también es aplicable a los datos de prueba.

En la **Figura 30** se muestra el código<sup>31</sup> utilizado para obtener la métrica de precisión, así como la matriz de confusión necesaria para calcular la tasa de error para este objetivo.

```
results = model.val(data="/content/Div2Datos/data.yaml", split="test")
# Mostrar los resultados
print(results)
```

**Figura 30.** Porción de código que obtiene respuestas sobre el rendimiento del dataset creado.

La ejecución del código de la **Figura 30** se muestra a continuación en la **Tabla 14**, donde se puede observar la evaluación del modelo YOLOv8n utilizando el subconjunto de datos de prueba (test) para la detección de sacos de balanceado de camarón. En total, se evaluaron 308 imágenes, conteniendo 1250 instancias de las tres clases objetivo: “Aquaxcel GROWER 5 (25 kg)”, “Aquaxcel Pre-Starter 2 (25 kg)”, y “ECOFEEED Purina (25 kg)”. El modelo alcanzó una precisión promedio (mAP) del 81.08% para umbrales de IoU entre 0.50 y 0.95; una precisión del 96.44% para un IoU de 0.50; y una precisión global del modelo de 89.02%.

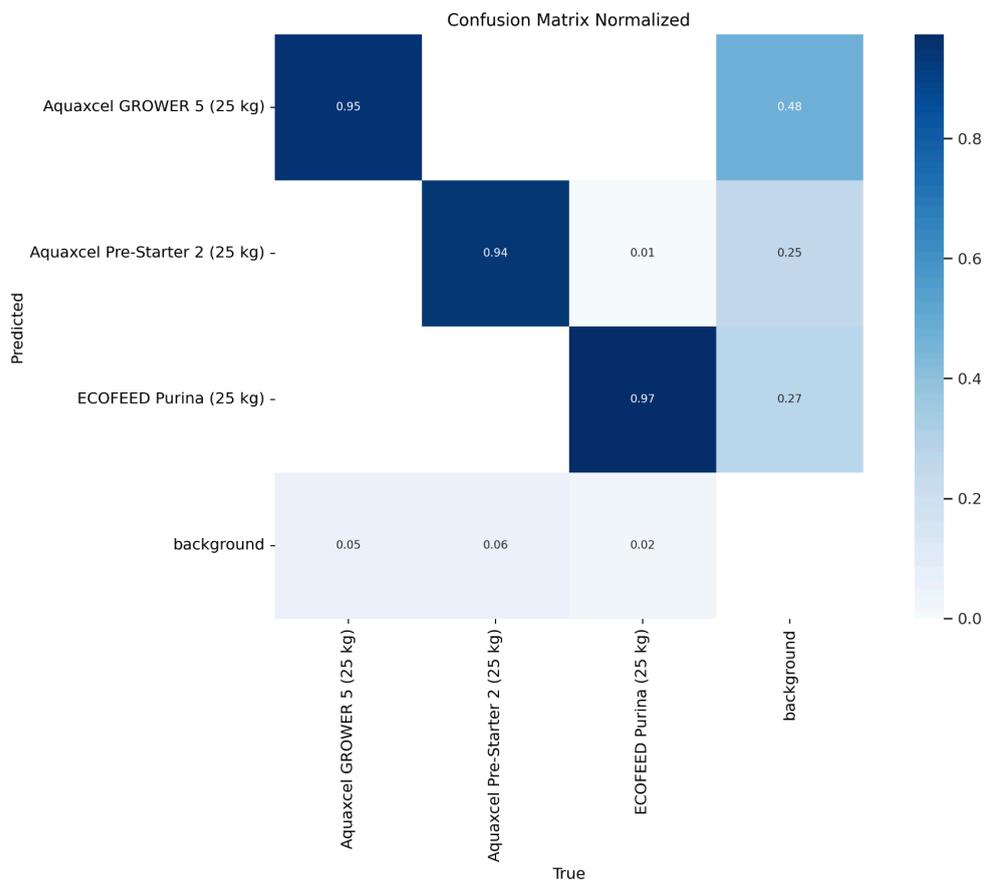
**Tabla 14.** Métricas de rendimiento del modelo YOLOv8n probado con el subconjunto de datos de prueba (test).

Métricas de Rendimiento del Modelo	
Precisión	0.8902
Loss	0.0188
Recall	0.9275
mAP50(B)	0.9644
mAP50-95(B)	0.8108

<sup>31</sup> [Código para sacar métricas del subconjunto de datos “Test” - Colab](#)

En el proceso de evaluación del modelo YOLOv8n, se utilizan varias métricas para medir su rendimiento, como la precisión (Precision) indica la proporción de predicciones verdaderas positivas entre todas las predicciones positivas realizadas por el modelo, reflejando cuántas de las predicciones del modelo son correctas; la pérdida (Loss) que tiene en cuenta componentes como la pérdida de caja (box\_loss), la pérdida de clasificación (cls\_loss) y la pérdida de regresión de distancia (dfl\_loss) mide cuán bien o mal está funcionando el modelo durante el entrenamiento que este caso fue el rendimiento del modelo con los datos de prueba; el Recall es la proporción de verdaderos positivos entre todos los elementos que deberían haber sido identificados como positivos, mostrando la capacidad del modelo para encontrar todas las instancias relevantes en los datos; la precisión media a IoU 0.50 (mAP50) mide la precisión media cuando se considera una Intersección sobre Unión (IoU) de 0.50, ósea que IoU realiza una división del área de superposición sobre el área de la unión de los cuadros delimitados, que quiere decir que toma el cuadro predicho y el cuadro real para sacar su exactitud, los umbrales de 0.50 quiere decir que alcanza el IoU promedio que significa igual o mayor a 0.50; mientras que la Precisión Media a IoU 0.50-0.95 (mAP50-95) es una métrica más estricta que mide la precisión media en un rango de IoU desde 0.50 hasta 0.95.

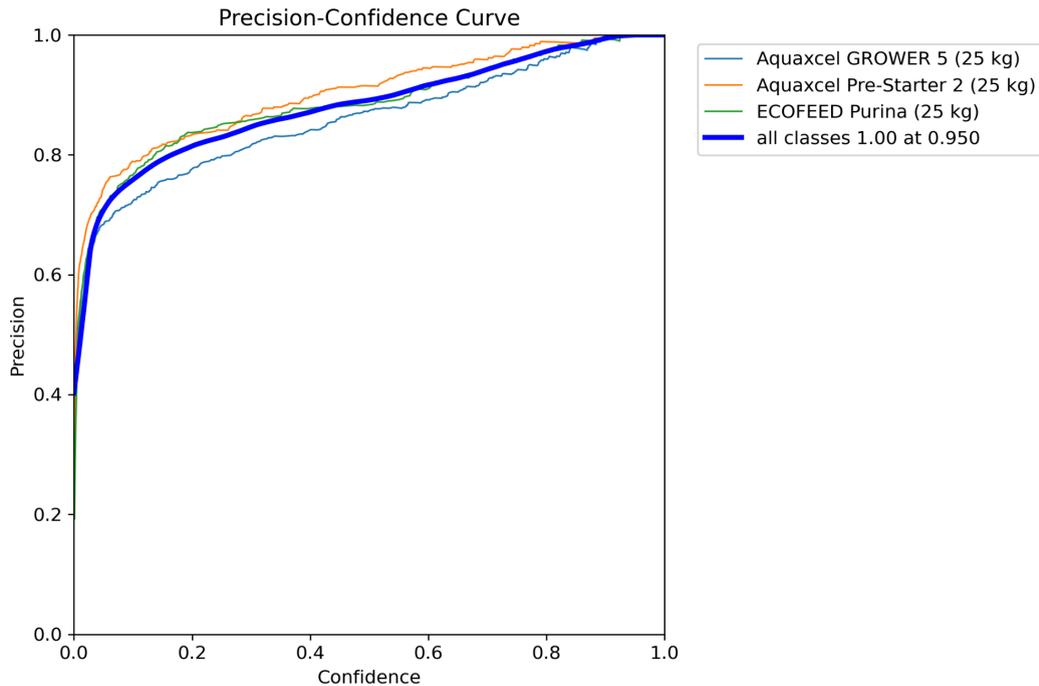
Con las métricas que salieron de la anterior ejecución, se muestra en la **Figura 31** una matriz de confusión normalizada para la clasificación de sacos de balanceado de camarón utilizando el modelo YOLOv8n, evaluado sobre el conjunto de datos de prueba. El modelo demostró una alta precisión en la detección de las clases “Aquaxcel GROWER 5 (25 kg)”, “Aquaxcel Pre-Starter 2 (25 kg)” y “ECOFEED Purina (25 kg)”, con valores de 95%, 94% y 97%, respectivamente. Sin embargo, se observó que una fracción considerable de las instancias de fondo fueron clasificadas incorrectamente como sacos de balanceado, con un 48% de las instancias de fondo etiquetadas como “Aquaxcel GROWER 5 (25 kg)”, un 25% como “Aquaxcel Pre-Starter 2 (25 kg)” y un 27% como “ECOFEED Purina (25 kg)”.



**Figura 31.** Matriz de confusión normalizada del rendimiento del mejor modelo para la clasificación de sacos de balanceado de camarón utilizando el modelo YOLOv8n.

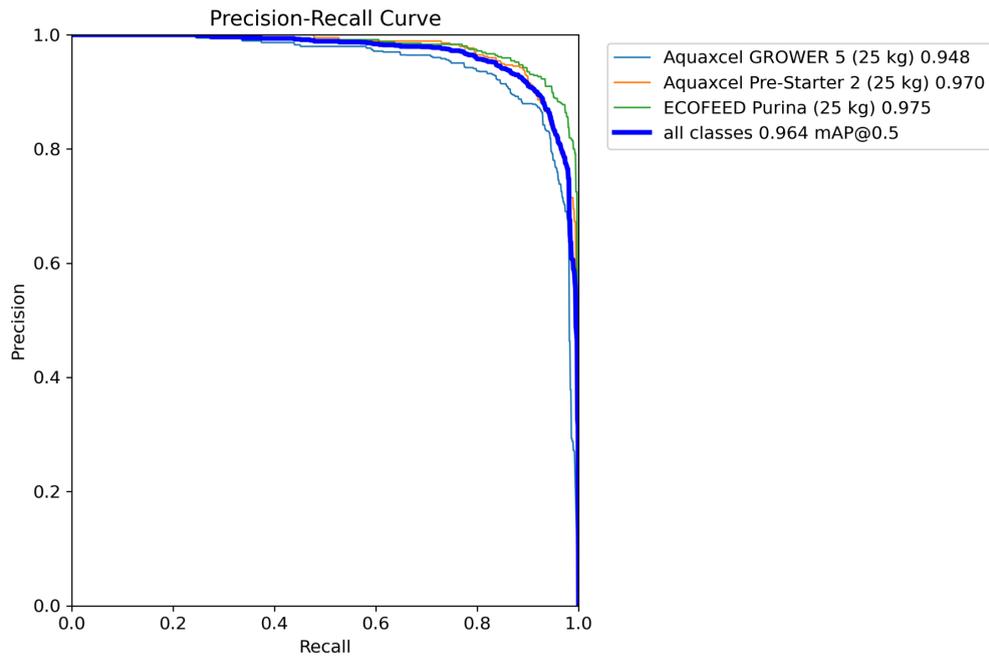
La **Figura 32** presenta la gráfica de la curva de precisión-confianza (Precision-Confidence Curve), muestra cómo varía la precisión en función de la confianza para cada una de las clases. Cada clase está representada por una línea de color diferente: la línea celeste para “Aquaxcel GROWER 5 (25 kg)”, la línea naranja para “Aquaxcel Pre-Starter 2 (25 kg)”, la línea verde para “ECOFEED Purina (25 kg)” y se incluye una línea de azul oscuro, para el comportamiento promedio de todas las clases combinadas, alcanzando una precisión de 1.00 en un nivel de confianza de 0.950.

La gráfica indica que, a medida que aumenta la confianza en las predicciones del modelo, la precisión también mejora, alcanzando valores cercanos a 1.0 para niveles altos de confianza.



**Figura 32.** Gráfica de Precision-Confidence Curve para la Clasificación de Sacos de Balanceado de Camarón utilizando el Modelo YOLOv8n.

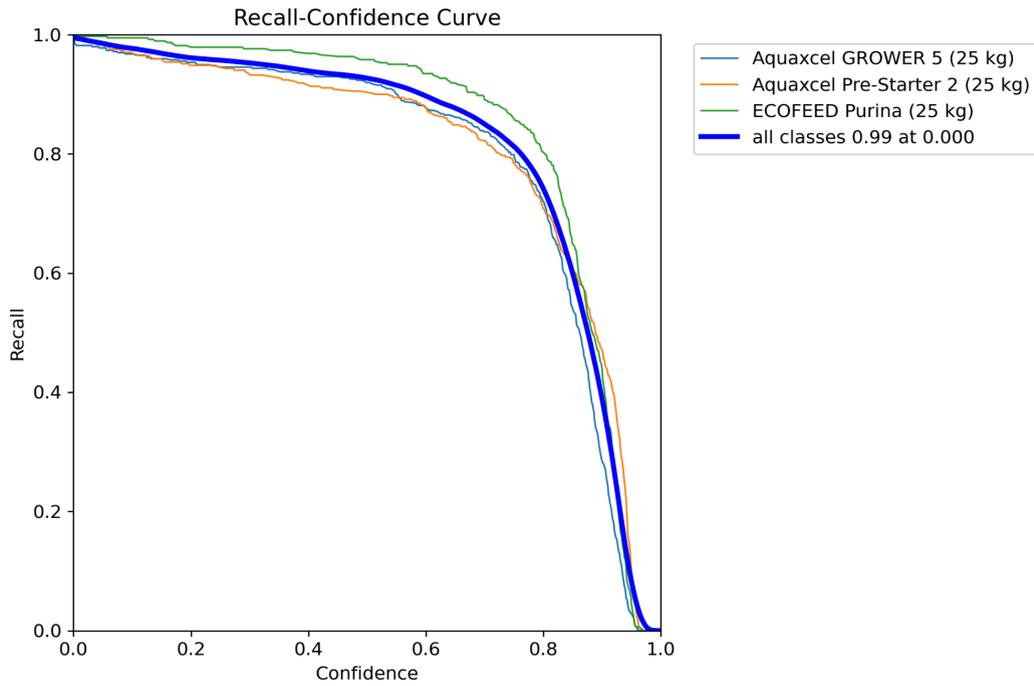
La gráfica de la curva de precisión-recall (Precision-Recall Curve) presentada en la **Figura 33**, muestra cómo varía la precisión en función de la recuperación para cada una de las clases, representada por una línea de color diferente: la línea celeste para “Aquaxcel GROWER 5 (25 kg)” con valor de 0.948, la línea naranja para “Aquaxcel Pre-Starter 2 (25 kg)” con valor de 0.970, la línea verde para “ECOFEED Purina (25 kg)” con valor de 0.975 y la línea azul oscuro, mostrando el comportamiento promedio de todas las clases combinadas, alcanzando un mAP (mean Average Precision) de 0.964 a un IoU de 0.5. La curva indica que el modelo mantiene una alta precisión, incluso con valores elevados de recuperación.



**Figura 33.** Gráfica de Precision-Recall Curve para la Clasificación de Sacos de Balanceado de Camarón utilizando el Modelo YOLOv8n.

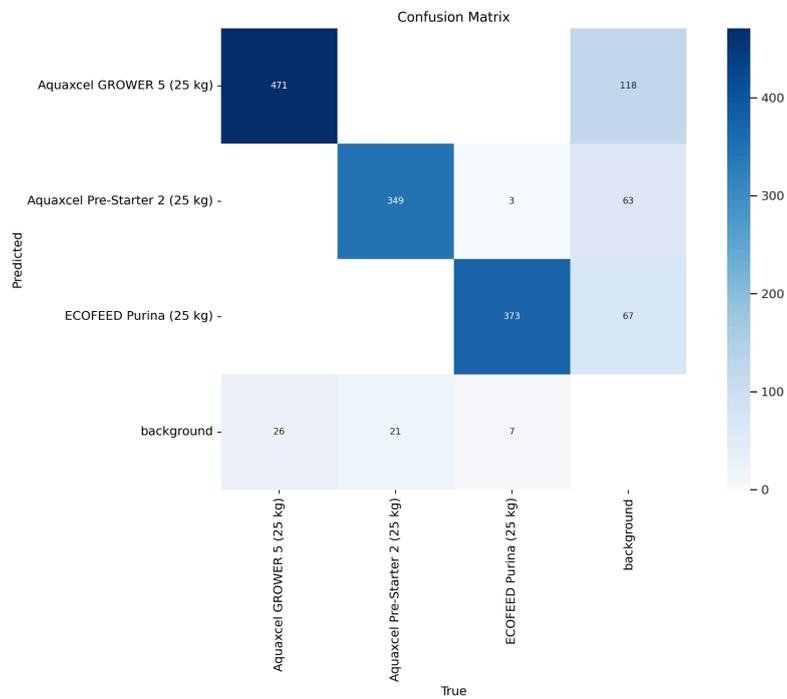
La **Figura 34** muestra la curva de recall-confianza (Recall-Confidence Curve), la gráfica revela cómo varía la recuperación en función de la confianza para cada una de las clases, y estas están representadas por una línea de color diferente: la línea celeste para “Aquaxcel GROWER 5 (25 kg)”, la línea naranja para “Aquaxcel Pre-Starter 2 (25 kg)”, la línea verde para “ECOFEED Purina (25 kg)” y adicionalmente, una línea azul oscuro muestra el comportamiento promedio de todas las clases combinadas, alcanzando una recuperación de 0.99 a un nivel de confianza de 0.000.

La curva indica que, a medida que aumenta la confianza en las predicciones del modelo, el recall disminuye, este comportamiento sugiere que el modelo YOLOv8n es capaz de detectar correctamente la mayoría de las instancias cuando la confianza es baja, pero la recuperación disminuye a medida que se incrementa la confianza, lo cual es un comportamiento esperado en modelos de detección de objetos.



**Figura 34.** Gráfica de Recall-Confidence Curve para la Clasificación de Sacos de Balanceado de Camarón utilizando el Modelo YOLOv8n.

Las métricas actuales, que salieron como resultado por defecto de YOLO, no proporcionan la métrica de “tasa de error”, por lo que se utilizaron los datos de la matriz de confusión de la **Figura 35** para realizar el cálculo manualmente.



**Figura 35.** Matriz de confusión no normalizado del rendimiento del mejor modelo para la clasificación de sacos de balanceado de camarón utilizando el modelo YOLOv8n.

Los valores de la matriz de confusión muestran parámetros con valores como Verdaderos Positivos (TP), Verdaderos Negativos (TN), Falsos Positivos (FP) y Falsos Negativos (FN) (ver **Tabla 15**), que son proporcionados por el rendimiento del modelo YOLOv8n, para cada clase y valores globales, estos parámetros son importantes, porque con ellos se pueden realizar los cálculos manuales para explorar el rendimiento del modelo, incluyendo la métrica de tasa de error.

**Tabla 15.** Valores de las métricas asociadas a la matriz de confusión del modelo YOLOv8n entrenado.

Métrica	Aquaxcel GROWER 5 (25 kg)	Aquaxcel Pre-Starter 2 (25 kg)	ECOFEED Purina (25 kg)	Valores Globales
TP	471	349	373	1193
TN	635	814	803	2252
FP	118	63	67	248
FN	26	24	7	57

*True Positive/Verdadero Positivo (TP); True negative/Verdadero Negativo (TN); False Positive/Falso Positivo (FP); False negative/Falso Negativo (FN).*

Con los valores proporcionados del rendimiento del modelo YOLOv8n del conjunto de datos de prueba se realiza el cálculo manual de la fórmula de tasa de error:

$$Taza\ de\ Error_{Aquaxcel\ GROWER\ 5\ (25\ kg)} = \frac{(FP + FN)}{(TP + TN + FP + FN)} = \frac{118+26}{471+635+118+26} = \frac{144}{1250} = 0.1152$$

$$Taza\ de\ Error_{Aquaxcel\ Pre-Starter\ 2\ (25\ kg)} = \frac{(FP + FN)}{(TP + TN + FP + FN)} = \frac{63+24}{349+814+63+24} = \frac{87}{1250} = 0.0696$$

$$Taza\ de\ Error_{ECOFEED\ Purina\ (25\ kg)} = \frac{(FP + FN)}{(TP + TN + FP + FN)} = \frac{67+7}{373+803+67+7} = \frac{74}{1250} = 0.0592$$

Realizando el cálculo en porcentajes se tiene las siguientes tasas de error por clase:

- Aquaxcel GROWER 5 (25 kg), la tasa de error es del 11.52%.
- Aquaxcel Pre-Starter 2 (25 kg), la tasa de error es del 6.96%.
- ECOFEED Purina (25 kg), la tasa de error es del 5.92%.

Tomando los datos globales, la tasa de error del modelo es la siguiente:

$$Taza\ de\ Error_{Global} = \frac{(FP + FN)}{(TP + TN + FP + FN)} = \frac{248+57}{1193+2252+248+57} = \frac{305}{3750} = 0.0813$$

Los resultados de la evaluación del conjunto de datos del mejor modelo entrenado muestra un 0.0813 del cálculo manual de la tasa de error, lo que significa que las predicciones realizadas por el modelo son incorrectas bajo esa cantidad.

## 7. Discusión

**7.1. Primer objetivo: Recolectar imágenes de sacos de balanceado de camarón Aquaxcel GROWER 5 (25 kg), Aquaxcel Pre-Starter 2 (25 kg) y ECOFEED Purina (25 kg) para anotarlas y formar un dataset diverso y representativo, tomando como referencia la metodología CRISP-DM y las prácticas de MLOps.**

Un conjunto de datos de imágenes debe ser representativa para poder ser aplicado en cualquier modelo de visión por computadora, tal como se ha demostrado en varios de los trabajos relacionados (ver **Tabla 1**), en este proyecto se siguieron como referencias las prácticas MLOps principalmente para el control de versiones, también la metodología CRISP-DM por sus primeras fases y las recomendaciones del maestro en conjuntos de datos, para poder asegurar la calidad del conjunto de datos creado, alineándose con procedimientos similares con los trabajos como **TR01**, **TR03**, **TR04**, **TR05**, **TR06** y **TR09**, en el cual se le da la importancia necesaria a la captura de los datos en diversos entornos como en condiciones, para poder garantizar su representatividad.

La tarea de la captura de imágenes se debe realizar en diversas condiciones y entornos, como se sugiere en los trabajos **TR01** y **TR03**, en el cual es un aspecto a tener en cuenta para mejorar la adaptabilidad del modelo en entornos complicados, sin embargo, en lugar de simplemente copiar estos pasos, se continuó con las recomendaciones del Ing. Oscar Cumbicus que se tomaron a lo largo de todo el proyecto, quien menciona la importancia de capturar imágenes con buena iluminación y diferentes ángulos, aunque se tomó en consideración la técnica del trabajo **TR03** para la captura de los datos, que se basa en la utilización de una cámara de teléfono móvil sin flash ni zoom, asegurando condiciones de luz natural, ayudando a asegurar que los datos cumplieran las condiciones de captura (ver **Tabla 3**), para conseguir 3280 imágenes; con ello se continuó con un procedimiento similar al **TR03**, que realiza un énfasis en la limpieza de datos y el uso de herramientas de anotación sencillas y precisas como Labellmg, utilizada también en trabajos como **TR04**, poseyendo al final 3145 datos anotados.

Tomando los trabajos de **TR05** y **TR06** se tomó la decisión de dividir el conjunto de datos en un 80% para entrenamiento, 10% para validación y 10% para prueba, en el cual también se basó en las prácticas MLOps que recomienda esta división, aunque otros trabajos recomiendan una división con una menor cantidad en el subconjunto de entrenamiento.

La mejora del conjunto de datos fue necesaria para construir un modelo más confiable, como se ha demostrado en trabajos como **TR03** y **TR06**, donde la mejora del conjunto de datos resultó en un rendimiento mejorado en términos de precisión. Aunque no se aplicaron técnicas de control de calidad por terceros como en **TR04**, se tomó a consideración la importancia de mantener la consistencia y precisión de los datos. Similarmente al **TR05**, se eliminó el desequilibrio en los datos mediante una evaluación manual a criterio del autor del presente documento TIC, eliminando aquellas que no cumplían con los criterios de calidad y conservando, 3075 datos. En nuestro proyecto, antes de la mejora del conjunto de datos, se obtuvo una precisión del 92% en “Aquaxcel GROWER 5 (25 kg)”, del 96% en “Aquaxcel Pre-Starter 2 (25 kg)” y del 97% en “ECOFEED Purina (25 kg)”. Después de la mejora, las precisiones aumentaron a 94% para la primera clase, 98% para la segunda y 98% para la tercera.

Aunque la precisión haya aumentado gracias a la mejora del conjunto de datos, los falsos positivos (FP) también se redujeron, sobre todo en la clase “Aquaxcel GROWER 5 (25 kg)” que paso de estar en 54% a un 47%, sigue siendo una cifra a tener en cuenta. Se hipotetiza que esto se debe a que la captura de imágenes en esta clase en particular es porque se tenían en mayor cantidad imágenes de sacos apilados, que por consecuencia se capturaban menos detalles por imagen. El cual se debe prestar más atención a este punto para tener mejor consistencia y equilibrio en los datos.

## **7.2. Segundo objetivo: Utilizar el modelo YOLOv8n con el dataset creado para medir la precisión y la tasa de error en la detección de sacos de balanceado de camarón Aquaxcel GROWER 5 (25 kg), Aquaxcel Pre-Starter 2 (25 kg), y ECOFEED Purina (25 kg).**

Con la utilización del modelo YOLOv8n para probar el conjunto de datos creado para la detección de productos acuícolas, demostró una precisión promedio (mAP) de 81.08% para umbrales de IoU entre 0.50 y 0.95, y de 96.44% en umbrales de IoU de 0.50, además de proporcionar la métrica global de precisión del modelo que fue de 89.02%, se comprobó la efectividad del modelo para detección de sacos de balanceado de camarón, considerando que los sacos tienen características similares, como el color blanco predominante, la tipografía, y también una presentación similar.

En cambio, la obtención de la métrica de la tasa de error del modelo que se obtuvo de forma manual es de 0.0813, lo que indica que existe un margen de mejora en la clasificación, especialmente en la reducción de falsos positivos, lo que demuestra que se puede confiar en el modelo entrenado, pero también apunta a la necesidad de mejorar la

discriminación entre los sacos y el fondo para reducir los falsos positivos, en particular, a la clase “Aquaxcel GROWER 5 (25 kg)” que tiene una tasa de error específica del 0.1152.

En términos de impacto, este proyecto presenta un comienzo para la creación de un conjunto de datos más robusta y con más clases, siguiendo los mismos pasos y tomando en cuenta los problemas y aciertos dentro del mismo; la aplicación en entornos industriales y sobre todo en el sector acuícola con sistemas de deep learning, pueden ayudar a la gestión de inventarios y logística como es el caso del **Anexo 2**, para disminuir los errores humanos y mejorando la eficiencia operativa, aunque el modelo se limita actualmente a solo tres productos, puede implementarse para aplicaciones específicas y trabajar con este conjunto de datos y modelo, para posteriores investigaciones y aplicaciones de deep learning, parecidos a los observados en el trabajo **TR08**, donde la aplicación de técnicas de aprendizaje profundo mejoró los procesos de control de calidad en la industria.

## 8. Conclusiones

- La creación del conjunto de datos de imágenes para a la detección de objetos especializados en productos acuícolas, como es el caso de los sacos de balanceado de camarón “Aquaxcel GROWER 5 (25 kg)”, “Aquaxcel Pre-Starter 2 (25 kg)”, y “ECOFEED Purina (25 kg)” evaluado y entrenado con el modelo YOLOv8n, alcanzo una precisión global del 89.02%, y en cuanto la tasa de error global llego a un 0.0813.
- La aplicación de la metodología de minería de datos CRISP-DM y las prácticas de MLOps que se tomaron como referencia, resultaron ser competentes a pesar de no estar creadas directamente para la creación de un conjunto de datos de imágenes, la recolección inicial se tomó con las condiciones de captura que dio como resultado 3280 datos, que pasaron por tareas que mejoraron el conjunto de datos, en el cual se puede destacar la aplicación de los criterios de inclusión y exclusión que se aplicaron en dos tareas, como es el caso de la tarea 2 y la 3 con el objetivo de brindar un filtro extra, para obtener un conjunto de datos representativo, diverso y bien organizado, pero al obtener el resultado inicial del rendimiento del modelo se pudo evidenciar un problema con el desequilibrio de las clases, mostrando la clase más afectada por su mayor cantidad de instancias una precisión individual del 92%, donde se necesitó una mejora al conjunto de datos, el cual se aplicó la eliminación de imágenes, dando como resultado un total del conjunto de datos de 3075 imágenes anotadas, con una mejora en su precisión en todas las clases de forma individual y general, destacando a “Aquaxcel GROWER 5 (25 kg)” que es la clase que contenía una mayor cantidad de instancias, con su precisión el cual subió hasta el 94%, demostrando que la calidad de los datos es una prioridad para poder utilizarlos en los modelos de deep learning.
- La utilización del modelo YOLOv8n con el conjunto de datos especializado para la detección de los sacos de balanceado de camarón pudo demostrar una precisión global de 89.02%, junto con una tasa de error global del 0.0813; en el caso individual las clases obtuvieron precisiones y tasas de error como 95% con 0.1152 para la clase “Aquaxcel GROWER 5 (25 kg)”, 94% con 0.0696 para la clase “Aquaxcel Pre-Starter 2 (25 kg)”, y un 97% junto con un 0.0592 para la clase “ECOFEED Purina (25 kg)”, estos resultados demuestra que pueden ser mejorados por una mayor diversidad en el conjunto de datos para reducir los falsos positivos del modelo.

## 9. Recomendaciones

- Es recomendable incrementar el conjunto de datos con más imágenes que capturen una mayor variedad de escenarios y condiciones de iluminación, con la inclusión de más datos que ayudará a mejorar la generalización del modelo y a reducir los sesgos que pueden surgir de un conjunto de datos limitado, como se observó en la clase “Aquaxcel GROWER 5 (25 kg)”, lo que permitirá que el modelo se adapte mejor a diferentes entornos.
- Se recomienda ampliar el conjunto de datos para incorporar otros productos (clases del modelo) relevantes en la industria acuícola, esto aumentará la aplicabilidad del modelo en diferentes contextos industriales.
- La recolección de datos se le debe de dar la importancia necesaria, ya sea con la combinación de técnicas de captura de datos junto con la definición de condiciones de captura, para que tengan la mejor calidad posible, evitando sesgos y lograr variabilidad tanto en entornos e iluminación, se puede seguir las pautas de la **TR09** para abarcar este punto.

### Limitaciones

- Actualmente, el conjunto de datos desarrollado tiene limitaciones en el momento que se realizan aplicaciones en los entornos de producción, por lo que se recomienda aplicarlo para la gestión de los productos de las clases definidas, y desarrollar modelos de deep learning para tareas específicas para estos mismos productos acuícolas.
- Si se tiene problemas con los recursos de hardware o son insuficientes para la ejecución de scripts que tengan que ver con entrenamientos de modelos de IA o procesamiento de datos, se puede utilizar Google Colab para estos tipos de dificultades.

### Trabajos futuros

- Se debe colaborar con personas que estén dentro de la industria acuícola y de la academia, para asegurar que el desarrollo y la aplicación del modelo sean efectivos, ya que estos ofrecen recomendaciones y retroalimentación muy importantes para la implementación práctica del modelo.
- Se recomienda ampliar el conjunto de datos con más productos (clases) para mejorar la variabilidad de las imágenes capturadas, lo que permitirá no solo la mejora de la precisión y la generalización del modelo, sino que también ampliarán su aplicabilidad a diferentes contextos en sectores industriales.

- Utilizar otros modelo de detección de objetos para realizar una comparación de rendimiento, y sacar conclusiones del mejor modelo para este conjunto de datos desarrollado.

## 10. Bibliografía

- [1] J. Arévalo, J. Orjuela, and R. Arias, “Innovación para potenciar al sector acuícola ecuatoriano,” *Ciencia Latina Revista Científica Multidisciplinar*, vol. 6, pp. 3451–3467, 11 2022.
- [2] N. Rane, “YOLO and Faster R-CNN object detection for smart Industry 4.0 and Industry 5.0: Applications, challenges, and opportunities”, *SSRN Electron. J.*, 2023. Accedido el 19 de junio de 2024. [En línea]. Disponible: <https://doi.org/10.2139/ssrn.4624206>
- [3] W. Vásquez-Quispesivana, M. Inga y I. Betalleluz-Pallardel, “Artificial intelligence in aquaculture: Basis, applications, and future perspectives”, *Sci. Agropecu.*, vol. 13, n.º 1, pp. 79–96, marzo de 2022. Accedido el 19 de junio de 2024. [En línea]. Disponible: <https://doi.org/10.17268/sci.agropecu.2022.008>
- [4] W. Jay Vanegas, M. G. Padilla Santamaria y M. K. Rodelo Molina, “Políticas públicas ante la revolución de la inteligencia artificial en Colombia”, *Rev. Venez. Gerencia*, vol. 29, n.º 106, pp. 865–883, marzo de 2024. Accedido el 19 de junio de 2024. [En línea]. Disponible: <https://doi.org/10.52080/rvgluz.29.106.26>
- [5] *Inteligencia artificial : 101 cosas que debes saber hoy sobre nuestro futuro - 1. edición.* Alienta Ed., 2018.
- [6] C. Janiesch, P. Zschech y K. Heinrich, “Machine learning and deep learning”, *Electron. Markets*, abril de 2021. Accedido el 19 de junio de 2024. [En línea]. Disponible: <https://doi.org/10.1007/s12525-021-00475-2>
- [7] N. Sharma, R. Sharma y N. Jindal, “Machine learning and deep learning applications-a vision”, *Global Transitions Proc.*, vol. 2, n.º 1, pp. 24–28, junio de 2021. Accedido el 19 de junio de 2024. [En línea]. Disponible: <https://doi.org/10.1016/j.gltp.2021.01.004>
- [8] A. I. Khan y S. Al-Habsi, “Machine learning in computer vision”, *Procedia Comput. Sci.*, vol. 167, pp. 1444–1451, 2020. Accedido el 19 de junio de 2024. [En línea]. Disponible: <https://doi.org/10.1016/j.procs.2020.03.355>
- [9] K. Y. Jiménez Cueva, “Determinación de la madurez de frutos del café mediante el reconocimiento de imágenes utilizando un Modelo Basado en Redes Neuronales Convolucionales”, Trabajo de Titulación, Univ. Nac. Loja, Loja, 2023.
- [10] G. Parés Marín, “Desarrollo de un sistema de detección de prendas de ropa en imágenes de moda mediante la implementación de redes YOLOv8 personalizadas”, Trabajo Fin de Grado, Univ. Politec. Valencia, Valencia, 2023.
- [11] E. M. Boderp, M. P. Lopez, A. E. Congacha, E. E. Cajamarca y C. H. Morales, “Google Collaboratory as an alternative for the processing of a convolutional neural network”, *Espacios*, vol. 41, n.º 7, p. 10, 2020.
- [12] M. Urretavizcaya, “Sistemas inteligentes en el ambito de la educacion”, *Inteligencia Artif.*, vol. 5, n.º 12, abril de 2001. Accedido el 19 de junio de 2024. [En línea]. Disponible: <https://doi.org/10.4114/ia.v5i12.703>
- [13] Y. Medina Carbó, C. M. Alonso Echevarría y I. Reinoso Castillo, “Intelligent system for the diagnosis of pests and diseases in rice cultivation”, *Dialnet*, vol. 12, n.º 4, p. 15, 2021.

- [14] V. H. Ayma Quirita, P. M. Achancaray Díaz, S. W. Arauco Canchumuni, y P. J. Soto Vega, «Desafíos del aprendizaje profundo en la visión por computador», CIIS, pp. 49-53, dic. 2022.
- [15] K. F. Moncayo Suárez. "Visión por computador para reconocimiento de malezas en cultivos de tomate riñón de invernadero, mediante redes neuronales," B.S. Thesis. , 2021. [Online]. Disponible en: <https://repositorio.utn.edu.ec/handle/123456789/11567L>.
- [16] A. Sánchez López, «Inteligencia artificial y visión por computadora aplicada a la educación», RO, vol. 3, n.º 2, pp. 61–73, jun. 2022.
- [17] M. A. Wani and S. Afzal, "A New Framework for Fine Tuning of Deep Networks," 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA), Cancun, Mexico, 2017, pp. 359-363, doi: 10.1109/ICMLA.2017.0-135.
- [18] J. Haotian, L. Dongyue y Z. Hongyang R, "Robust fine-tuning of deep neural networks with hessian-based generalization guarantees", en Proc. 39th Int. Conf. Mach. Learn., Maryland, Estados Unidos, 3 de junio de 2022. Boston MA: Northeastern Univ., 2022, p. 31.
- [19] J. M. Peña Lorenzo, "Aplicación de técnicas de aprendizaje profundo (deep learning) para la detección de objetos en industria 4.0", Trabajo Fin de Máster, Esc. Tec. Super. Ing. Telecomunicación, Valladolid, 2020.
- [20] F. Rojas Ramos, "Detección de objetos usando cámaras y sensores LIDAR", Tesis de maestría, Benemérita Universidad Autónoma de Puebla, 2020. . citation.available: <https://hdl.handle.net/20.500.12371/9884>. citation.accessed: Jun. 19, 2024
- [21] Amit, Y., Felzenszwalb, P., Girshick, R. (2021). Object Detection. In: Ikeuchi, K. (eds) Computer Vision. Springer, Cham. [https://doi.org/10.1007/978-3-030-63416-2\\_660](https://doi.org/10.1007/978-3-030-63416-2_660)
- [22] [1] C. S. R. Moreno et al, "Viabilidad de las redes neuronales en la detección de objetos," Revista Ibérica De Sistemas e Tecnologias De Informação, pp. 981-1000, 2020/04//. Available: <https://www.proquest.com/scholarly-journals/viabilidad-de-las-redes-neuronales-en-la/docview/2388304882/se-2>.
- [23] A. Paullada, I. D. Raji, E. M. Bender, E. Denton y A. Hanna, "Data and its (dis)contents: A survey of dataset development and use in machine learning research", Patterns, vol. 2, n.º 11, p. 100336, noviembre de 2021. Accedido el 19 de junio de 2024. [En línea]. Disponible: <https://doi.org/10.1016/j.patter.2021.100336>
- [24] J. J. Espinosa Zúñiga, "Aplicación de metodología CRISP-DM para segmentación geográfica de una base de datos pública", Ing. Investig. Tecnol., vol. 21, n.º 1, pp. 1–13, enero de 2020. Accedido el 19 de junio de 2024. [En línea]. Disponible: <https://doi.org/10.22201/fi.25940732e.2020.21n1.008>
- [25] F. Santana Falcón, "Análisis de datos abiertos de fútbol aplicando la metodología CRISP-DM", Trabajo Fin de Máster, Univ. Politec. Madr. Esc. Tec. Super. Ing. Telecomunicacion, Madrid, 2023.
- [26] M. R. Marcial Rodríguez, A. G. López Montaña, L. Rodríguez Páez, R. Rico Molina y D. Gutiérrez Cruz, "Aplicación de minería de datos mediante la metodología CRISP-DM para defunciones accidentales y violentas en Nezahualcóyotl", Numero Espec. Rev. Aristas Cienc. Basica Apl., vol. 10, n.º 18, p. 6, 2023.
- [27] M. Santamaría Sacristán, "Machine Learning Operations (MLOps): Contexto actual y tendencias futuras", Trabajo Fin de Máster, Univ. Rioja, Rioja, 2023.

- [28] I. Núñez Molina, “Desarrollo de una herramienta MLOps de apoyo a proyectos de ciencia de datos”, Trabajo Fin de Grado, Univ. Politec. Valencia, Valencia, 2023.
- [29] J. Ramas, “Implementación de las prácticas de MLOps para PATE”, Trabajo de Maestría, Univ. ORT Uruguay, Montevideo, 2022.
- [30] S. R. Mallqui Herrada, “Google drive y la gestión documentaria en instituciones educativas de la red 5, UGEL 05, 2022.”, Trabajo Fin de Grado, Univ. Cesar Vallejo, Lima, 2023.
- [31] E. Cervera, R. Marin y J. Marin, “Más allá de jupyter: Usando google colab para la programación de robots”, en *Jorn. Autom.* (43<sup>a</sup>. 2022. Logrono), Castellón de la Plana, España.
- [32] H. G. . Gil Novoa y D. S. . Hernández Gómez, «G DRIVE COMO REPOSITORIO EN PROYECTOS TECNOACADEMIA», *Con-ciencia y Tec.*, vol. 4, n.º 1, pp. 129–132, feb. 2022.
- [33] M. Kuroki, “Using Python and Google Colab to teach undergraduate microeconomic theory”, *Int. Rev. Econ. Educ.*, vol. 38, p. 100225, noviembre de 2021. Accedido el 20 de junio de 2024. [En línea]. Disponible: <https://doi.org/10.1016/j.iree.2021.100225>
- [34] L. Tan, T. Huangfu, L. Wu y W. Chen, “Comparison of RetinaNet, SSD, and YOLO v3 for real-time pill identification”, *BMC Med. Inform. Decis. Making*, vol. 21, n.º 1, noviembre de 2021. Accedido el 20 de junio de 2024. [En línea]. Disponible: <https://doi.org/10.1186/s12911-021-01691-8>
- [35] J. Tan, Y. Chen y S. Jiao, “Visual studio code in introductory computer science course: An experience report”, en *Conf.*, July 2017, Washington, Estados Unidos.
- [36] M. Plainer, “Study of visual studio code”, 1 de diciembre de 2020, Practical Course — Contributing to an Open-Source Project, New York, Estados Unidos. [En línea]. Disponible: <https://www.proof.cit.tum.de/teaching/osp/WS20/assets/pr-plainer-vscode.pdf>
- [37] J. Raraz-Vidal, «La Importancia de las Bases de Datos para el Entrenamiento en Inteligencia Artificial», *Rev Peru Investig Salud*, vol. 7, n.º 3, pp. 121–122, sep. 2023.
- [38] J. Coelho, M. T. Valente, L. Milen y L. L. Silva, “Is this GitHub project maintained? Measuring the level of maintenance activity of open-source projects”, *Inf. Softw. Technol.*, vol. 122, p. 106274, junio de 2020. Accedido el 20 de junio de 2024. [En línea]. Disponible: <https://doi.org/10.1016/j.infsof.2020.106274>
- [39] N. Zöllner, J. H. Morgan y T. Schröder, “A topology of groups: What GitHub can tell us about online collaboration”, *Technolog. Forecasting Social Change*, vol. 161, p. 120291, diciembre de 2020. Accedido el 20 de junio de 2024. [En línea]. Disponible: <https://doi.org/10.1016/j.techfore.2020.120291>
- [40] J. Llerena Izquierdo, *Codifica en python*. Cuenca: Univ. Politec. Sales., 2020.
- [41] “The Python Tutorial”, 3.12.4 Documentation » The Python Tutorial. [En línea]. Disponible: <https://docs.python.org/3/tutorial/>
- [42] T. Domínguez Mínguez, *Visión artificial: Aplicaciones prácticas con OpenCV - Python*. MARCOMBO, S. L., 2021. [En línea]. Disponible: [https://books.google.com.mx/books?hl=es&lr=&id=FE1OEAAAQBAJ&oi=fnd&pg=PT5&dq=OpenCV&ots=3zykGm1h-Z&sig=zxwo1ZhvYku\\_nQBbh6M1NijPqK0#v=onepage&q=OpenCV&f=false](https://books.google.com.mx/books?hl=es&lr=&id=FE1OEAAAQBAJ&oi=fnd&pg=PT5&dq=OpenCV&ots=3zykGm1h-Z&sig=zxwo1ZhvYku_nQBbh6M1NijPqK0#v=onepage&q=OpenCV&f=false)
- [43] A. Sharma, J. Pathak, M. Prakash and J. N. Singh, "Object Detection using OpenCV and Python," 2021 3rd International Conference on Advances in Computing,

- Communication Control and Networking (ICAC3N), Greater Noida, India, 2021, pp. 501-505, doi: 10.1109/ICAC3N53548.2021.9725638.
- [44] R. TH. Hasan and A. . Bibo Sallow, "Face Detection and Recognition Using OpenCV", *jscdm*, vol. 2, no. 2, pp. 86–97, Oct. 2021, Accessed: Jun. 20, 2024. [Online]. Available: <https://publisher.uthm.edu.my/ojs/index.php/jscdm/article/view/8791>
- [45] T. Chen and Q. Zeng, "Research on Bubble Detection Based on Improved YOLOv8n," in *IEEE Access*, vol. 12, pp. 9659-9668, 2024, doi: 10.1109/ACCESS.2024.3353196.
- [46] Q. Liu et al., "DSW-YOLOv8n: A New Underwater Target Detection Algorithm Based on Improved YOLOv8n", *Electronics*, vol. 12, n.º 18, p. 3892, septiembre de 2023. Accedido el 24 de junio de 2024. [En línea]. Disponible: <https://doi.org/10.3390/electronics12183892>
- [47] Z. Wang, Z. Hua, Y. Wen, S. Zhang, X. Xu y H. Song, "E-YOLO: Recognition of estrus cow based on improved yolov8n model", *Expert Syst. With Appl.*, p. 122212, octubre de 2023. Accedido el 24 de junio de 2024. [En línea]. Disponible: <https://doi.org/10.1016/j.eswa.2023.122212>
- [48] S. Kumari, A. Gautam, S. Basak and N. Saxena, "YOLOv8 Based Deep Learning Method for Potholes Detection," 2023 IEEE International Conference on Computer Vision and Machine Intelligence (CVMI), Gwalior, India, 2023, pp. 1-6, doi: 10.1109/CVMI59935.2023.10465038.
- [49] M. N. Uddin, M. Sakibul Islam Sakib, S. Nawer and R. T. Mohona, "Improved Fire Detection by YOLOv8 and YOLOv5 to Enhance Fire Safety," 2023 26th International Conference on Computer and Information Technology (ICCIT), Cox's Bazar, Bangladesh, 2023, pp. 1-6, doi: 10.1109/ICCIT60459.2023.10441615.
- [50] Q. Luo, C. Wu, G. Wu and W. Li, "A Small Target Strawberry Recognition Method Based on Improved YOLOv8n Model," in *IEEE Access*, vol. 12, pp. 14987-14995, 2024, doi: 10.1109/ACCESS.2024.3356869.
- [51] S. Lin, L. Huang, X. Liu, G. Chen y Z. Fu, "A construction waste landfill dataset of two districts in Beijing, China from high resolution satellite images", *Sci. Data*, vol. 11, n.º 1, abril de 2024. Accedido el 24 de junio de 2024. [En línea]. Disponible: <https://doi.org/10.1038/s41597-024-03240-0>. Lin, L. Huang, X. Liu, G. Chen y Z. Fu, "A construction waste landfill dataset of two districts in Beijing, China from high resolution satellite images", *Sci. Data*, vol. 11, n.º 1, abril de 2024. Accedido el 24 de junio de 2024. [En línea]. Disponible: <https://doi.org/10.1038/s41597-024-03240-0>
- [52] M. Yuan et al., "A dataset for fine-grained seed recognition", *Sci. Data*, vol. 11, n.º 1, abril de 2024. Accedido el 24 de junio de 2024. [En línea]. Disponible: <https://doi.org/10.1038/s41597-024-03176-5>
- [53] V. Anand, S. Gupta, D. Koundal, W. Y. Alghamdi y B. M. Alsharbi, "Deep learning-based image annotation for leukocyte segmentation and classification of blood cell morphology", *BMC Med. Imag.*, vol. 24, n.º 1, abril de 2024. Accedido el 24 de junio de 2024. [En línea]. Disponible: <https://doi.org/10.1186/s12880-024-01254-z>
- [54] A. N. Yumang, J. Anthony S. Katipunan y A. L. Galido, "Reconocimiento y recuento de materiales utilizados en restaurantes de comida rápida basados en la detección de objetos YOLO para el sistema de inventario", 2022 IEEE 14th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM), Isla de Boracay, Filipinas, 2022, pp. 1-6, doi: 10.1109/HNICEM57413.2022.10109440.

- [55] V. E.K. y C. Ramachandran, "Real-time Gender Identification from Face Images using you only look once (yolo)", 2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)(48184), Tirunelveli, India, 2020, pp. 1074-1077, doi: 10.1109/ICOEI48184.2020.9142989
- [56] B. Tong y M. Zhang, "Comparación de algoritmos de la serie YOLO en la detección de máscaras", Taller Internacional sobre Sistemas Inteligentes (IWIS) de 2023, Ulsan, Corea, República de, 2023, pp. 1-5, doi: 10.1109/IWIS58789.2023.10284631.

## 11. Anexos

**Anexo 1.** Entrevista sobre el desarrollo de conjuntos de datos para la detección de objetos.

**Título:** Entrevista sobre el desarrollo de conjuntos de datos para detección de objetos.

**Fecha:** 31/01/2024

**Presentación:** Esta entrevista es realizada con el propósito de adquirir una retroalimentación y recomendaciones sobre el tema de la creación del conjunto de datos de imágenes para realizar la detección y clasificación de objetos, con el objetivo de realizar un cambio a mejor en los procesos logísticos en las industrias mediante un modelo de machine learning, y que este sirva como un punto de partida para soluciones tecnológicas aplicadas en entornos de producción.

### **Entrevistador:**

- **Nombre:** Keyner Alexis Manchay Montoya
- **Rol:** Estudiante de la Carrera de Ingeniería en Ciencias de la Computación
- **e-mail:** keyner.manchay@unl.edu.ec
- **Teléfono:** 0991869497

### **Entrevistado:**

- **Nombre:** Oscar Miguel Cumbicus Pineda
- **Rol:** Docente de Machine Learning
- **Departamento o Área de Especialización:** Carrera de Computación / Facultad de la Energía, las Industrias y los Recursos Naturales no Renovables

### **Preguntas:**

#### **1. ¿Cómo describiría la importancia de los datasets en el campo de la detección de objetos mediante visión por computadora?**

Los datos es lo más importante dentro de machine learning, si no hay datos no se pueden hacer ni los modelos de inteligencia artificial o de machine learning, por lo tanto, la importante o la más alta importancia son los datos, en este caso los datasets. Si no tenemos un dataset para hacer visión por computadora, entonces no podemos utilizar los modelos, si no tenemos datasets de texto, no podemos hacer procesamiento de Lenguaje de natural, si no tenemos datasets de sonido, entonces no podemos hacer modelos de reconocimiento de voz. Por eso los datasets son lo más importante dentro del machine learning.

## 2. ¿Qué considera que son los componentes esenciales de un buen dataset para detección de objetos?

Un buen dataset debe tener un número adecuado de muestras dependientemente del problema; que si es visión es imagen, si es procesamiento, lenguaje natural es texto, si es voz es audio. Un número considerable de muestra deben abarcar la problemática o los individuos o la gran mayoría de individuos que están dentro del tema. Por ejemplo si es para un dataset de animales, las muestras debería estar distribuidas de manera equitativa, si es para reconocimiento de texto para 5 textos con la misma cantidad, entonces para que un dataset sea de buena calidad debe dividir los elementos que van a hacer parte de problema de machine learning deben tener una muestra representativa e igualitaria, todos deben tener, si es posible la misma cantidad de imagen.

**Representativa:** Que deba estar todo. Sí, hay 3 animales, bueno que todos estén en el dataset.

## 3. Al diseñar un dataset para detección de objetos, ¿qué características deben tener las imágenes y las anotaciones?

Deben de ser de alta calidad, en este caso deberían depender de la arquitectura que vayan a consumir, si la arquitectura de red neuronal dice que necesitan 400 x 400, pues deberían de tener esas características, entre más grande sean los tamaños de las fotos más demorara el procesamiento, porque en una foto grande más característica tiene que tomar, que una que tenga menos tamaño. En la gran mayoría se suele trabajar de 256 x 256.

- ¿Y si no hay un enfoque a una arquitectura específica, debería estar en alta calidad? Si, no hay ningún problema, pueden ser de alta calidad, pero se deberían tomar en cuenta los trabajos relacionados, es decir, revisar cuál es el tamaño de las fotos, que se utilizaron en trabajos similares a los tuyos, por ejemplo, para segmentación de objetos se necesitan 64 x 64 según los trabajos relacionados, entonces debería ser así, si es para detección entonces se deben ver trabajos para detección para ver el promedio de las imágenes, todo depende de la arquitectura que lo vaya a consumir.

## 4. ¿Cuál es el proceso típico que sigue para recopilar y verificar datos para un nuevo dataset de detección de objetos?

Primero se debe realizar la toma de las imágenes, justamente la toma de las imágenes deben ser, no solamente que las imágenes estén... por ejemplo, con bastante luz, también se deben tomar con poca luz, es decir, se deben tomar en varios contextos, con poca luz, con mucha luz, con un fondo blanco, con un fondo que existan personas, con un fondo donde haya animales, con diferentes fondos. Porque es el que va a permitir reconocer cuando existan este tipo de fondos, el algoritmo vaya a reconocer, ósea que tienes que tomar una foto se debe de tomar en diferentes circunstancias. También a distancia de un metro, a distancia de dos metros, dependiendo de las circunstancias que se pueda tener en ese momento, ese es el primer proceso.

Una vez que se tiene las fotografías, se debe determinar cuál de esas fotografías son válidas para el dataset. Una vez que se hayan apartado las imágenes válidas para crear tu dataset, con esas imágenes dependiendo de que sea de clasificación, detección o segmentación. Si es en clasificación se debe especificar de tal quintal tipo

uno, quintal tipo dos, se debe meter imágenes de cuando existe y cuando no existe, entonces se debe clasificar las imágenes de acuerdo con el tema que tengas.

- Tomar las imágenes
- Tomar las válidas
- Luego clasificarlas según la clase

**5. ¿Qué herramientas y tecnologías recomienda para la anotación y etiquetado de datasets destinados a la detección de objetos?**

Existen muchas herramientas, desde python puro, sacar características de Opencv, también hay roboflow, imageslabel que es una herramienta que también puede hacer etiquetados, labelme, que con este hicieron la tesis de Karla Jimenez, se graduó, hizo la tesis de la detección del producto del café.

**6. ¿Cómo se abordan los desafíos de la variabilidad y la ocultación parcial de objetos en imágenes?**

Hay algunas técnicas que deberías revisar justamente para poder renderizar las fotos, que quiere decir que tú puedes usar técnicas para que quites el fondo, aumentes la visibilidad del objeto que quieres detectar, esas son técnicas de renderizado, tú puedes quitar ruido de una foto, por ejemplo si se toman fotos con un solo fondo, es ahí donde se aplican estas técnicas, para poderle subir el brillo, cambiar el fondo, para eso se usan estas técnicas para poder mejorar las fotos que es renderizado. Lo óptimo es tomar las fotos, con luz clara, otro día con menos luz, etc. Y si no se consigue eso, se aplican estas técnicas. Lo ideal es conseguir fotos en todas las condiciones, por eso sé utilizar las técnicas de renderizado, si no consigues los diferentes fondos, pues es ahí donde se aplican el cambio de fondos, fondo de flores, fondo de personas, no vas a conseguir con diversos fondos, de hecho tú tienes que aplicar una técnica y ponerle, también darle la vuelta, entonces como hago eso, aplicando técnicas de renderizado o transformación de imágenes.

**7. ¿Cómo se valida la creación de un dataset de Detección de Objetos?**

Una de las formas de validar es hacer un test AB, ósea es darle una foto a un experto y que te diga, de que objeto se trata y que tú sepas con anterioridad qué objeto es. Por ejemplo, si tú le das una foto de un quintal A, tú ya sabes que es el quintal A, y el experto te dice que es el quintal A, entonces está corroborado que está bien, pero si él te dice que es el quintal B, entonces deberías revisar. Otra vez es coger el dataset ya etiquetado y utilizar una arquitectura de clasificación o un modelo de clasificación, ósea tú lo entrenas y el modelo lo clasificará en paquete A, B y C, cuantos de esas fotos fueron clasificadas como A, si de 10 fotos 9 las clasifico como B, entonces se debería revisar esas fotos, para ver si tiene más ruido, si fue tomada en condiciones más bajas, entonces puedes validar de esa forma.

## **Anexo 2.** Entrevista aplicada a empleada de la empresa ASOCAM.

**Título:** Entrevista sobre la gestión de inventario y la necesidad de la automatización en los procesos dentro de una empresa

**Fecha:** 05/02/2024

**Presentación:** Esta entrevista es realizada con el objetivo de entender como funcionan los procesos manuales dentro de industrias relacionadas con las zonas pesqueras, ya sea en sus productos como en sus funciones, para poder identificar posibles ineficiencias manuales dentro de la empresa, para ofrecer alguna posible solución, para una implementación futura, para que los procesos manuales tiendan a la automatización que cumplan con las necesidades de la empresa.

### **Entrevistador:**

- **Nombre:** Keyner Alexis Manchay Montoya
- **Rol:** Estudiante de la Carrera de Ingeniería en Ciencias de la Computación
- **e-mail:** keyner.manchay@unl.edu.ec
- **Teléfono:** 0991869497

### **Entrevistado:**

- **Nombre:** Marjorie Aponte Suárez
- **Rol:** Encargada de la contabilidad en la empresa
- **Departamento:** Contable y Administrativo

### **Preguntas:**

- 1. ¿Podría describir cómo se manejan los procesos logísticos en ASOCAM, especialmente en términos de organización, almacenamiento y distribución de los productos?**

Si, por ejemplo, cuando hacen un pedido a las chicas de abajo, que son las personas encargadas de realizar las ventas, entonces cuando hay mercadería, ellas revisan que todo esté bien y los van almacenando según el sector, por ejemplo, en la parte de abajo saben donde va el balanceado, lo que son los insumos, las bacterias, las vitaminas, ya lo tienen segmentado por secciones a los productos, también mantenemos otras bodegas donde contienen los carbonatos, lo que son mercadería bastante, entonces se almacenan en otras bodegas, aquí en Huaquillas, Hualtaco.

- 2. En el contexto de su operación logística, ¿cuáles son los principales desafíos o ineficiencias que han observado?**

Aquí si hay bastante, sobre todo antes ocurría en el tema de despachos, por ejemplo se equivocan en los productos, es que existen balanceados que se parecen, pero no son iguales, y a veces hay personal nuevo en bodega, los estibadores por ejemplo si abajo ellos facturan 100 sacos de un producto, y despachan otro producto que casi tienen el mismo nombre, pero no es el mismo; en lo que son los carbonatos también hay diferentes carbonatos con malla 200 y malla 100, entonces también existe ese error al momento de despachar, ósea de entregar otro producto. Otro error es que se entrega demás, al menos eso más antes ocurría, ahora hay un poco más de control

**3. ¿Podría compartir algún ejemplo específico de situaciones donde se han enfrentado a dificultades en la gestión de su inventario o en la cadena de suministro?**

Las personas mismas que generan las ventas, las de oficina, tienen que salir haya en el producto o a veces en el muelle cuando es formal, y cuando es terrestre afuera, por lo que son ellas mismas las que tienen que ir a contar saco por saco, no se fían mucho de los estibadores, ya que hay control de que se entregan de más o se entrega de menos, por eso el proceso de conteo es manual por parte del departamento de venta. Y también los anteriormente dichos.

**4. Hasta ahora, ¿han implementado alguna tecnología o sistema automatizado para apoyar sus procesos logísticos?**

Si es así, ¿qué tipo de tecnología y cómo ha impactado en su operación? No, como le digo se toman fotos cuando se despacha, entonces si por ejemplo, digamos que al cuadrar el inventario, se ve que faltan productos, entonces van a la foto para ver si de repente se fue demás o si hay algún inconveniente que el cliente diga no "Mira me llevo 5 sacos menos" entonces van a la foto y de esa manera se puede ver cuantos son los productos que se pueden observar, talvez en su momento no se pueda ver; es como el balanceado, hay personas que llevan 500 sacos de balanceado, y eso es imposible en una foto ver, pero sí cuando llevan insumos, pues esas son cosas que sí se pueden contar fácilmente.

**5. ¿Cuál es su opinión sobre la implementación de tecnologías de automatización e inteligencia artificial en la logística? ¿Considera que podrían ser beneficiosas para ASOCAM?**

Si, en algún momento se hablaba de que incluso cada producto tenga su propio código de barra, para que se pueda ver incluso la fecha de caducidad, pero claro imagine que son bastantísimos productos y si pasa siempre que se caducan productos y que no es rápido observarlo por las personas encargadas, porque como le digo, no se lleva un control del tema, que quizás en este caso si sería factible, hubiera algo, o un programa, que si ellos escanean un código, este quede en un programa y que después este les esté mostrando cuando vaya a caducar con anticipación, para que de esta manera se pueda realizar una promoción del mismo y no como es en el caso que llega a perdida. porque ya sea a dado cuenta cuando el producto ya ha caducado y no cuando el producto este por caducar. También por tema de stock, aquí también se está contando a dedo, cuantos productos hay, porque como le digo estar repartidos los productos por segmentos, entonces si un programa por ejemplo que diga que este lote hay tanto balanceado, como que le vaya arrojando un stock, porque igual nosotros si tenemos un sistema contable, que nos muestra cuanto stock tiene, si por ejemplo tenga 1000 sacos de balanceado se vendió 500, pues quedo 500, pero abajo es otra cosa, porque el cliente compro los 500 sacos, pero el cliente va llevando de 100 en 100, ósea que se haga un conteo de inventario, abajo en bodega va a haber más que acá en el sistema contable, por eso hace falta un control interno o un sistema que notifique, porque siempre están con ese inconveniente.

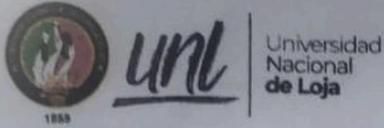
**6. Si tuviera la oportunidad de mejorar los procesos logísticos con una solución tecnológica, ¿qué aspectos le gustaría que esta abordara o mejorara?**

Sí, los anteriormente dichos.

- 7. En términos de tecnologías de automatización, ¿está familiarizado con el concepto de datasets especializados para entrenamiento de sistemas inteligentes? ¿Cree que un dataset específico para sacos podría ser útil en su contexto?**

No habíamos tenido conocimiento.

**Anexo 3.** Permiso de autorización para captura de fotografía dentro de las instalaciones de la empresa ASOCAM.



Loja, 05 de febrero del 2024

Ing.

**Marjorie Aponte Suárez**

**Departamento Contable y Administrativo en Asociación de Productores Camaroneros**

De mi consideración:

Yo, **Keyner Alexis Manchay Montoya**, con el número de cédula **0706844255**, estudiante de Ingeniería en Computación en la Universidad Nacional de Loja, solicito comedidamente a usted el permiso para poder tomar fotografías dentro de las instalaciones de la empresa ASOCAM, específicamente a los productos empaquetados en sacos. El propósito de mi solicitud es obtener autorización para utilizar estas imágenes en el desarrollo de un proyecto con fines académicos de la carrera de computación.

Agradezco de antemano su disposición y apoyo a la investigación académica.

Atentamente,

Keyner Alexis Manchay Montoya  
C.I.: 0706844255  
Institución: Universidad Nacional de Loja  
Correo: keyner.manchay@unl.edu.ec  
Celular: 0991869497

n.



**Anexo 4.** Certificado de la traducción del resumen del proyecto TIC.



**UNL**

Universidad  
Nacional  
de Loja

Loja, 01 de agosto de 2024

Lic. Pedro Geovanny Calva Jiménez

**LICENCIADO EN PEDAGOGÍA DEL IDIOMA INGLÉS**

**CERTIFICO:**

Que el resumen del Trabajo de Integración Curricular cuyo título es: **Desarrollo de un Dataset Especializado para la Detección Automática de Sacos de Balanceado de Camarón Aquaxcel GROWER 5 (25 kg), Aquaxcel Pre-Starter 2 (25 kg) y ECOFEED Purina (25 kg) utilizando el Modelo YOLOv8**, del aspirante **Keyner Alexis Manchay Montoya**, con cédula de identidad Nro. **0706844255**, de la Carrera de Computación de la Universidad Nacional de Loja, ha sido traducido al inglés y cumple con las características propias del idioma extranjero.

Lo certifico en honor a la verdad y autorizo hacer uso del presente en lo que a sus intereses convenga.

Lo certifico en honor a la verdad.

Lic. Pedro Geovanny Calva Jiménez

**1150428496**

Nro. Reg. Senecyt: 1031-2022-2421774

**LICENCIADO EN PEDAGOGÍA DEL IDIOMA INGLÉS**