



Esta obra está bajo una

[Licencia Creative Commons](https://creativecommons.org/licenses/by/4.0/)

[Atribución - 4.0 Internacional \(CC BY 4.0\)](https://creativecommons.org/licenses/by/4.0/)

Vea una copia de esta licencia en

<https://creativecommons.org/licenses/by/4.0/deed.es>





FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

Tesis

Pesaje no invasivo de ganado vacuno mediante modelos híbridos basado en segmentación de imágenes y regresión

Para optar el título profesional de Ingeniero de Sistemas e Informática

Autor:

Héctor Luis Fernández Rosillo
<https://orcid.org/0009-0002-3750-888X>

Asesor:

Ing. Dr. Miguel Ángel Valles Coral
<https://orcid.org/0000-0002-8806-2892>

Tarapoto, Perú

2026



FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

Tesis

Pesaje no invasivo de ganado vacuno mediante modelos híbridos basado en segmentación de imágenes y regresión

Para optar el título profesional de Ingeniero de Sistemas e Informática

Autor:

Héctor Luis Fernández Rosillo

Sustentado y aprobado el 17 de marzo del 2026, ante el honorable jurado:

Presidente de Jurado
Ing. Dr. Carlos Enrique López
Rodríguez

Secretario de Jurado
Ing. Dr. Jorge Damian
Valverde Iparraguirre

Vocal de Jurado
Ing. Dr. Janina Cotrina Linares
de Quezada

Asesor
Ing. Dr. Miguel Ángel Valles
Coral

Tarapoto, Perú

2026



**ACTA DE SUSTENTACIÓN
PARA OPTAR EL TÍTULO DE INGENIERO DE SISTEMAS E INFORMÁTICA**

Resolución N° 010-2026-UNSM/FISI-D (16.03.2026)

FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA – ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

A las 11:00 horas del día martes, 17 de marzo del año 2026, se inició el acto público de sustentación de la tesis titulada: PESAJE NO INVASIVO DE GANADO VACUNO MEDIANTE MODELOS HÍBRIDOS BASADO EN SEGMENTACIÓN DE IMÁGENES Y REGRESIÓN, presentado por el Bach. HÉCTOR LUIS FERNÁNDEZ ROSILLO, con el Asesor: Ing. Dr. Miguel Ángel Valles Coral.

Instalado los miembros de jurado calificador conformado por:

Presidente : Ing. Dr. Carlos Enrique López Rodríguez

Secretario : Ing. Dr. Jorge Damian Valverde Iparraguirre

Vocal : Ing. Dra. Janina Cotrina Linares de Quezada

El presidente del jurado dirigió brevemente unas palabras y a continuación el secretario dio lectura a la Resolución N° 010-2026-UNSM/FISI-D.

Seguidamente el autor expuso el trabajo de investigación y el jurado realizó las preguntas pertinentes, respondidas por el sustentante y eventualmente por el asesor, con la venia del jurado.

Una vez terminada la ronda de preguntas el jurado procedió a deliberar para determinar la calificación final, para lo cual dispuso un receso de quince (15) minutos, con participación del asesor con voz, pero sin voto y sin la presencia del sustentante y otros participantes del acto público.

Luego de aplicar los criterios de calificación con estricta observancia del principio de objetividad y de acuerdo con los puntajes en escala vigesimal (de 0 a 20), según el Anexo 4.2. del RG-CTI, la nota de sustentación otorgada resultante del promedio aritmético de los calificativos emitidos por cada uno de los miembros del jurado fue ... *dieciséis* (16).

De acuerdo con el Artículo 40° del RG – CTI, la nota obtenida es *aprobado* y correspondiente a la calificación de ... *muy bueno* ...; leído este resultado en presencia de todos los participantes del acto de sustentación, el secretario dio lectura a las observaciones subsanables al informe final que el autor deberá corregir y alcanzar al jurado en un plazo máximo de treinta (30) días calendario.



Universidad Nacional de San Martín
Facultad de Ingeniería de Sistema e Informática
Ciudad Universitaria - Jr. Amorarca # 315 - Morales



Firman los integrantes del jurado calificador, asesor y el autor de la tesis en señal de conformidad, dando por concluido el acto a las ...12.30... horas, el mismo día 17 de marzo del 2026.

Ing. Dr. Carlos Enrique López Rodríguez
Presidente

Ing. Dr. Jorge Damian Valverde Iparraguirre
Secretario

Ing. Dra. Janina Cotrina Linares de Quezada
Vocal

Ing. Dr. Miguel Angel Valles Coral
Asesor

Héctor Luis Fernández Rosillo
Autor

Constancia de asesoramiento

Quien suscribe el presente documento,

HACE CONSTAR:

Que, habiendo acompañado en la ejecución de la tesis titulada: Pesaje no invasivo de ganado vacuno mediante modelos híbridos basado en segmentación de imágenes y regresión. Elaborado por el tesista:

Bachiller en Ingeniería de Sistemas e Informática: **Héctor Luis Fernández Rosillo.**

Por lo que doy conformidad para los trámites correspondientes, dejo como constancia el presente documento y firmo.

Tarapoto, 17 de marzo de 2026



.....
Ing. Dr. Miguel Ángel Valles Coral
Asesor

Declaratoria de autenticidad

Héctor Luis Fernández Rosillo, con DNI N° 75166773, bachiller de la Facultad de Ingeniería de Sistemas e Informática de la Universidad Nacional de San Martín, autor de la tesis titulada: **Pesaje no invasivo de ganado vacuno mediante modelos híbridos basado en segmentación de imágenes y regresión.**

Declaramos bajo juramento que:

1. La tesis presentada es de autoría propia.
2. La redacción fue realizada respetando las citas y referencia de las fuentes bibliográficas consultadas, siguiendo las normas APA actuales.
3. Toda información que contiene la tesis no ha sido plagiada.
4. Los datos presentados en los resultados son reales, no han sido alterados ni copiados, por tanto, la información de esta investigación debe considerarse como aporte a la realidad investigada.

Por lo antes mencionado, asumo bajo responsabilidad las consecuencias que deriven de mi accionar, sometiéndome a las leyes de nuestro país y normas vigentes de la Universidad Nacional de San Martín.

Tarapoto, 17 de marzo de 2026.



Héctor Luis Fernández Rosillo

DNI N° 75166773

Ficha de identificación

<p>Título: Pesaje no invasivo de ganado vacuno mediante modelos híbridos basado en segmentación de imágenes y regresión</p>	<p>Área de investigación: Ingeniería y Tecnología Línea de investigación: Ciencias de la computación Sublínea de investigación: Inteligencia artificial y recuperación de información Grupo de investigación: Grupo de Investigación IA (RESOLUCIÓN N° 134–2021–UNSM/FISI/CFT) Tipo de investigación: Básica <input type="checkbox"/>, Aplicada <input checked="" type="checkbox"/>, Desarrollo experimental <input type="checkbox"/></p>
<p>Autor: Héctor Luis Fernández Rosillo</p>	<p>Facultad de Ingeniería de Sistemas e Informática Escuela Profesional de Ingeniería de Sistemas e Informática https://orcid.org/0009-0002-3750-888X</p>
<p>Asesor: Ing. Dr. Miguel Ángel Valles Coral</p>	<p>Dependencia local de soporte: Facultad de Ingeniería de Sistemas e Informática Escuela Profesional de Ingeniería de Sistemas e Informática Unidad o Laboratorio Ingeniería de Sistemas e Informática https://orcid.org/0000-0002-8806-2892</p>

Dedicatoria

Dedico esta investigación al Sr. Dany Cristian, propietario del fundo donde se desarrolló este estudio, por su generosidad al permitir que este trabajo se llevara a cabo en su propiedad, lo que hizo posible la realización de este proyecto. También quiero dedicar este logro a mis padres, quienes han sido mi mayor apoyo en cada etapa de mi vida, ya que su amor, confianza y fortaleza me han acompañado siempre, brindándome la motivación necesaria para alcanzar esta meta.

Agradecimientos

Este trabajo no hubiera sido posible sin el apoyo y la orientación de muchas personas. Mi más sincero agradecimiento al Ing. Dr. Miguel Ángel Valles Coral, mi asesor, por brindarme su sabiduría y acompañarme en este proceso, ayudándome a fortalecer mi enfoque crítico e investigativo. A mi familia, que siempre estuvo presente, brindándome su apoyo, comprensión y aliento, lo cual fue fundamental para llegar hasta aquí. A mis amigos cercanos, por ser una fuente constante de motivación y por sus palabras de aliento que me impulsaron a no rendirme. También, agradezco a la Universidad Nacional de San Martín, que me brindó la oportunidad de formarme profesionalmente, y a todos los docentes que, con su conocimiento, contribuyeron al desarrollo de mis competencias y crecimiento académico.

Índice general

Ficha de identificación.....	6
Dedicatoria.....	7
Agradecimientos	8
Índice general.....	9
Índice de tablas	11
Índice de figuras.....	12
RESUMEN	13
ABSTRACT	14
CAPÍTULO I INTRODUCCIÓN A LA INVESTIGACIÓN	15
CAPÍTULO II MARCO TEÓRICO.....	19
2.1. Antecedentes de la investigación.....	19
2.2. Fundamentos teóricos.....	20
2.2.1. Estimación del peso del ganado vacuno.....	20
2.2.2. Modelos híbridos basados en segmentación de imágenes y regresión ..	22
CAPÍTULO III MATERIALES Y MÉTODOS	26
3.1. Ámbito y condiciones de la investigación	26
3.1.1. Contexto de la investigación	26
3.1.2. Periodo de ejecución	26
3.1.3. Autorizaciones y permisos	27
3.1.4. Control ambiental y protocolos de bioseguridad	27
3.1.5. Aplicación de principios éticos internacionales	27
3.2. Sistema de variables.....	27
3.2.1. Variables principales.....	27
3.2.2. Variables secundarias	28
3.3. Procedimientos de la investigación	28

3.3.1. Actividades del objetivo específico 1: Construir un conjunto de datos mediante la toma de imágenes de ganado vacuno asociadas a su peso real en un centro de pesaje.	30
3.3.2. Actividades del objetivo específico 2: Desarrollar modelos híbridos basados en segmentación de imágenes y regresión para estimar de forma no invasiva el peso del ganado vacuno.	35
3.3.3. Actividades del objetivo específico 3: Evaluar las métricas de rendimiento de los modelos híbridos basados en segmentación de imágenes y regresión en la estimación del peso del ganado vacuno.	41
CAPÍTULO IV RESULTADOS Y DISCUSIÓN.....	45
4.1. Resultado específico 1: Construir un conjunto de datos mediante la toma de imágenes de ganado vacuno asociadas a su peso real en un centro de pesaje. ...	45
4.2. Resultado específico 2: Desarrollar modelos híbridos basados en segmentación de imágenes y regresión para estimar de forma no invasiva el peso del ganado vacuno.	49
4.2.1. Modelo híbrido YOLOv8 + Random Forest.....	54
4.2.2. Modelo híbrido YOLOv8 + XGBoost	55
4.2.3. Modelo híbrido YOLOv8 + LightGBM	57
4.3. Resultado específico 3: métricas de rendimiento de los modelos híbridos ...	58
CONCLUSIONES	65
RECOMENDACIONES	66
REFERENCIAS BIBLIOGRÁFICAS	67
ANEXOS	73
Anexo 1: Autorización de acceso a fondo.....	73
Anexo 2: Código de desarrollo de modelos desarrollados	74

Índice de tablas

Tabla 1. Descripción de variables por objetivo específico 1	28
Tabla 2. Descripción de variables por objetivo específico 2	28
Tabla 3. Métricas según 16 iteraciones	59
Tabla 4. Prueba de homogeneidad de varianza	61
Tabla 5. Resultados de la prueba ANOVA	61
Tabla 6. Prueba de Tukey para comparación del RMSE entre modelos híbridos.....	62
Tabla 7. Estadísticos descriptivos	62

Índice de figuras

Figura 1. Ubicación de la investigación	26
Figura 2. Diagrama de etapas del objetivo específico 1	30
Figura 3. Adecuación del entorno físico para la recolección de datos	31
Figura 4. Registro del peso real del ganado	32
Figura 5. Colocación del objeto de referencia	33
Figura 6. Captura de imágenes del perfil del ganado vacuno.....	34
Figura 7. Imágenes nombradas con código y peso real de gana ejemplar.....	35
Figura 8. Diagrama de etapas del objetivo específico 2	36
Figura 9. Estructura del dataset en Google Drive.....	37
Figura 10. Diagrama de etapas del objetivo específico 3	42
Figura 11. Pasarela de conducción del ganado hacia la balanza	45
Figura 12. Ejemplar bovino durante el registro de peso en balanza electrónica.....	46
Figura 13. Círculo de vinil colocado para calibración geométrica de imágenes.....	47
Figura 14. Imagen del ejemplar en vista lateral para análisis morfológico.....	47
Figura 15. Estructura del dataset organizado en Google Drive	48
Figura 16. Detección y normalización del círculo de referencia.....	50
Figura 17. Segmentación del ganado (YOLOv8 preentrenado).....	52
Figura 18. Extracción de características geométricas	53
Figura 19. Evaluación gráfica del modelo híbrido YOLOv8 + Random Forest.....	55
Figura 20. Evaluación gráfica del modelo híbrido YOLOv8 + XGBoost	56
Figura 21. Evaluación gráfica del modelo híbrido YOLOv8 + LightGBM Regressor	57
Figura 22. Gráficos de medias	63

RESUMEN

Pesaje no invasivo de ganado vacuno mediante modelos híbridos basado en segmentación de imágenes y regresión

La presente investigación se desarrolló con el propósito de comparar modelos híbridos basados en segmentación de imágenes y regresión para la estimación no invasiva del peso de ganado vacuno en la provincia de San Martín, Perú. Esta iniciativa surge ante la persistencia de métodos tradicionales de pesaje que generan estrés animal, implican altos costos y presentan baja accesibilidad en zonas rurales. El estudio se ejecutó entre enero y julio del 2025 en el fundo "Ganadería DR" del distrito de Zapatero, y consistió en la construcción de un dataset compuesto por aproximadamente 3500 imágenes de perfil de bovinos, cada una asociada a su peso real registrado en balanza electrónica. Se empleó un diseño experimental factorial 3x1, utilizando YOLOv8 para la segmentación automática de imágenes y tres algoritmos de regresión (XGBoost, Random Forest y LightGBM) para la estimación del peso a partir de características morfológicas extraídas computacionalmente. La evaluación de los modelos se realizó mediante validación cruzada (k-fold = 5) y métricas como MAE, RMSE y R², cuyos resultados fueron comparados aplicando ANOVA. Los hallazgos evidencian diferencias significativas entre modelos, destacando el modelo YOLOv8 + XGBoost por su menor error promedio y mayor capacidad explicativa. Se concluye que los modelos híbridos propuestos constituyen una alternativa efectiva, precisa y accesible para estimar el peso del ganado sin contacto físico, aportando soluciones innovadoras para la modernización de la ganadería en contextos rurales.

Palabras clave: visión computacional, aprendizaje automático, bienestar animal, procesamiento de imágenes, modelos predictivos.

ABSTRACT

Non-invasive weighing of cattle using hybrid models based on image segmentation and regression

This study was conducted to compare hybrid models based on image segmentation and regression for the non-invasive estimation of cattle weight in the province of San Martín, Peru. This initiative arose in response to the continued use of traditional weighing methods, which cause animal stress, entail high costs, and are difficult to access in rural areas. The study was conducted between January and July 2025 at the “Ganadería DR” ranch in the district of Zapatero and consisted of building a dataset comprising approximately 3,500 profile images of cattle, each associated with its actual weight recorded on an electronic scale. A 3x1 factorial experimental design was employed, using YOLOv8 for automatic image segmentation and three regression algorithms (XGBoost, Random Forest, and LightGBM) to estimate weight based on computationally extracted morphological features. The models were evaluated using cross-validation (k-fold = 5) and metrics such as MAE, RMSE, and R^2 , with the results compared using ANOVA. The findings reveal significant differences between models, with the YOLOv8 + XGBoost model standing out due to its lower average error and greater explanatory power. It is concluded that the proposed hybrid models constitute an effective, accurate, and accessible alternative for estimating livestock weight without physical contact, providing innovative solutions for the modernization of livestock farming in rural contexts.

Keywords: computer vision, machine learning, animal welfare, image processing, predictive models.



CAPÍTULO I INTRODUCCIÓN A LA INVESTIGACIÓN

Cada año, en todo el mundo, se examinan más de mil millones de cabezas de ganado con el fin de determinar su valor comercial, lo que convierte al control del peso corporal en un indicador clave dentro de la industria ganadera (Ruchay et al., 2022). El peso vivo de los animales representa un factor esencial no solo para la comercialización, sino también para la salud y el manejo productivo de las unidades de explotación ganadera. En este contexto, el uso del puntaje de condición corporal ha permitido evaluar de forma objetiva el estado nutricional y sanitario del ganado (Zhao et al., 2023).

Con los avances tecnológicos, investigaciones recientes demuestran que los algoritmos de aprendizaje automático pueden predecir el peso vivo del ganado a partir de sus características morfológicas, lo cual ofrece una alternativa precisa y no invasiva frente a los métodos tradicionales de pesaje (Bhoj et al., 2022; Ruchay et al., 2022). A nivel internacional, se impulsa el uso de visión artificial e inteligencia artificial para optimizar el pesaje de ganado sin contacto físico, priorizando el bienestar animal, la eficiencia operativa y la sostenibilidad en la producción (Devi et al., 2024; Z. Liu et al., 2023; Ogawa et al., 2024). No obstante, en diversas regiones rurales del mundo, los métodos tradicionales aún prevalecen, generando pérdidas de eficiencia y afectando negativamente la calidad del manejo ganadero (Hou et al., 2023; Xiong et al., 2023).

En el ámbito latinoamericano, los avances tecnológicos aplicados a la ganadería enfrentan retos similares. Aunque países como Brasil, Argentina y México han comenzado a incorporar sistemas de visión por computadora y aprendizaje automático para mejorar la productividad bovina, existen todavía notables brechas en la adopción de estas tecnologías por parte de los pequeños y medianos productores. Estas brechas se explican por las restricciones económicas, culturales y de infraestructura (Jiang et al., 2024; Nguyen et al., 2023; Temenos et al., 2024).

En el contexto peruano, la actividad ganadera posee un alto potencial, especialmente en regiones tropicales como San Martín. Sin embargo, la realidad evidencia que la estimación del peso del ganado continúa realizándose mediante técnicas empíricas o instrumentos costosos de difícil acceso, particularmente para los productores de menor escala (Canaza-Cayo et al., 2024). La modernización del sector ganadero avanza de forma lenta, debido a la carencia de tecnologías adaptadas, la limitada formación técnica de los ganaderos y los elevados costos de las balanzas industriales (KOJIMA et al., 2022; Lozano, 2020).

En el trópico peruano, la implementación de tecnologías como la visión artificial resulta aún incipiente. Aunque se reconoce su potencial para mejorar la rentabilidad y la eficiencia del control de peso, las soluciones disponibles no han sido lo suficientemente ajustadas al entorno rural ni a las condiciones socioeconómicas locales (Paudel et al., 2023; Xu et al., 2024). En la provincia de San Martín, la mayoría de los ganaderos continúan estimando el peso del ganado vacuno mediante la simple observación visual, ante la falta de balanzas adecuadas, asequibles y accesibles (Lozano, 2020; Ormeño-Ayala & Zapata-Ttito, 2024). Esta práctica empírica incrementa los márgenes de error en decisiones relacionadas con la alimentación, la venta y el manejo sanitario, lo que se traduce en menores ingresos y mayores riesgos para la sostenibilidad de las unidades productivas familiares.

Diversos estudios destacan que los métodos tradicionales de pesaje no solo implican una elevada inversión en infraestructura y logística, sino que también inducen altos niveles de estrés en los animales, reduciendo su bienestar y afectando su productividad (Brando & Norman, 2023; Chen et al., 2023; G. Li et al., 2022; J. Zhang et al., 2021). Por ello, el uso de herramientas de visión artificial, mediante técnicas de segmentación de imágenes y modelos de regresión, se plantea como una solución innovadora, ética y económicamente viable para la estimación del peso del ganado en esta zona.

A pesar de los avances internacionales en sistemas automatizados de pesaje, en la provincia de San Martín aún no se han desarrollado investigaciones locales que validen la eficacia de modelos híbridos específicamente diseñados para el ganado vacuno. Este vacío científico limita la implementación de tecnologías basadas en evidencia, restringiendo el avance del sector ganadero regional (Bhoj et al., 2022). Además, la captura de imágenes en ambientes naturales plantea desafíos técnicos, debido al comportamiento impredecible del ganado y a las variaciones lumínicas del entorno rural (Peng et al., 2024).

Frente a esta problemática, esta investigación propone el desarrollo y evaluación de modelos híbridos que integran técnicas de segmentación de imágenes y algoritmos de regresión, orientados a la estimación no invasiva del peso de ganado vacuno en la provincia de San Martín. Este proyecto busca reducir la brecha tecnológica, incrementar la eficiencia de las prácticas ganaderas locales y promover métodos productivos que prioricen el bienestar animal (Arshad et al., 2024; G. Li et al., 2021; Xie et al., 2024).

Por consiguiente, la problemática abordada en este trabajo fue identificar las diferencias en el desempeño de los modelos híbridos basados en segmentación de imágenes y regresión para el pesaje no invasivo de ganado vacuno. Para responder a esta cuestión

general, se formularon tres interrogantes específicas que orientaron el desarrollo de la investigación: en primer lugar, se evaluó la adecuación del conjunto de datos generado mediante la toma de imágenes de ganado vacuno asociadas a su peso real, con el fin de determinar su utilidad para el entrenamiento de modelos híbridos; en segundo lugar, se analizó qué modelos híbridos, combinando técnicas de segmentación de imágenes y algoritmos de regresión, resultaban más apropiados para estimar de manera no invasiva el peso del ganado vacuno; y se examinó qué métricas de rendimiento permitían evaluar con mayor precisión la eficacia de los modelos desarrollados en función de su exactitud y eficiencia operativa.

Con el propósito de responder a estas interrogantes, se planteó la hipótesis general de que existían diferencias significativas en el desempeño de los modelos híbridos basados en segmentación de imágenes y regresión para el pesaje no invasivo de ganado vacuno. A partir de esta hipótesis general, se formularon tres hipótesis específicas que guiaron el desarrollo experimental de la investigación. La primera sostuvo que la construcción de un conjunto de datos mediante la toma de imágenes de ganado vacuno, asociadas a su peso real, permitió entrenar modelos híbridos con datos adecuados para estimaciones no invasivas. La segunda propuso que el desarrollo de modelos híbridos, integrando técnicas de segmentación de imágenes como YOLOv8 con algoritmos de regresión, fue efectivo para estimar el peso del ganado vacuno a partir de las características morfológicas extraídas. La tercera hipótesis estableció que las métricas de rendimiento como el Error Absoluto Medio (MAE), la Raíz del Error Cuadrático Medio (RMSE) y el coeficiente de determinación (R^2) permitieron evaluar con precisión la eficacia de los distintos modelos híbridos empleados en la estimación del peso del ganado vacuno.

En lo que respecta a los objetivos del estudio, se estableció como objetivo general comparar modelos híbridos basados en segmentación de imágenes y regresión para el pesaje no invasivo de ganado vacuno. Para ello, se desarrollaron tres objetivos específicos. En primer lugar, se construyó un conjunto de datos a partir de la toma de imágenes de vacas en un centro de pesaje, donde se empleó una pasarela para guiarlas hacia una balanza calibrada y se colocó un objeto de referencia circular en la zona abdominal para facilitar la captura estandarizada de su perfil. En segundo lugar, se desarrollaron modelos híbridos utilizando YOLOv8 para la segmentación de imágenes, un algoritmo de extracción de características morfológicas (área rellena, perímetro y bounding box), y modelos de regresión como XGBoost Regressor, Random Forest Regressor y LightGBM Regressor para estimar el peso. Luego, se evaluó el rendimiento

de los modelos aplicando métricas como MAE, RMSE y R^2 , y se compararon los resultados mediante ANOVA para determinar su precisión y eficiencia.

A partir del problema identificado y los objetivos definidos, el presente estudio fue estructurado en cuatro capítulos. En el Capítulo I se presentó la investigación, describiendo la problemática abordada, los objetivos propuestos y la hipótesis que orientó el desarrollo del trabajo. El Capítulo II desarrolló el marco teórico, en el cual se revisaron antecedentes relevantes y se expusieron los fundamentos conceptuales que sustentan la investigación. En el Capítulo III se expusieron los materiales utilizados, así como los métodos aplicados para la obtención de datos, la implementación de los modelos híbridos y su respectiva evaluación. El Capítulo IV mostró los resultados alcanzados junto con su respectivo análisis e interpretación. El informe concluye con las secciones de conclusiones, recomendaciones, referencias bibliográficas y anexos, que complementan el trabajo desarrollado.

CAPÍTULO II MARCO TEÓRICO

2.1. Antecedentes de la investigación

En la investigación de Ruchay et al. (2022), el objetivo fue desarrollar un método no invasivo para estimar el peso vivo del ganado bovino mediante visión por computadora, con el fin de optimizar la selección genética, el control sanitario y el momento del sacrificio. Para ello, implementaron un modelo de aprendizaje profundo utilizando imágenes RGB-D capturadas con cámaras Microsoft Kinect v2. Los autores evaluaron tres modelos predictivos, alcanzando una precisión del 91.6%, y concluyeron que su propuesta reduce el estrés animal y mejora la eficiencia del manejo ganadero. Este antecedente respalda el uso de tecnologías de visión por computadora para el pesaje no invasivo en bovinos.

En Sichuan, China. Peng et al. (2024) tuvieron como objetivo estimar en tiempo real el peso de novillas de yak bajo condiciones climáticas extremas y comportamiento indómito, mediante un modelo automático basado en YOLOv8. Utilizaron imágenes 2D para detectar la postura e identificar a cada animal individualmente, logrando un 81.8% de precisión en postura y 97.8% en identificación. Concluyeron que esta tecnología es eficaz y reduce significativamente el estrés animal. Este estudio justifica el uso de modelos basados en aprendizaje profundo para entornos difíciles y con especies de difícil manejo.

En China. Hou et al. (2023) plantearon como objetivo desarrollar un modelo de predicción no invasiva del peso vivo del ganado bovino mediante aprendizaje profundo tridimensional. Utilizaron la arquitectura PointNet++ con datos capturados por sensores LiDAR, que generaron nubes de puntos 3D del contorno del animal. Su modelo alcanzó una precisión del 95.1% y un error absoluto medio de 3.2%. El objetivo de este estudio fue reducir la labor manual y el estrés causado por el pesaje tradicional, aportando evidencia sobre la efectividad de tecnologías tridimensionales aplicadas al sector pecuario.

La investigación de Xiong et al. (2023), tuvo como objetivo predecir el peso corporal y clasificar las puntuaciones de condición corporal (BCS) en vacas de carne adultas en Estados Unidos, mediante el uso de imágenes de profundidad de bajo costo. Su enfoque se basó en estimar el volumen corporal utilizando técnicas de aprendizaje automático. Con datos de 58 vacas cruzadas, lograron una fuerte correlación entre el volumen

proyectado y el peso corporal, destacando al modelo de árboles agrupados como el más eficaz. Este estudio evidencia que el volumen corporal es una variable clave para la predicción no invasiva del peso en ganado vacuno.

En Perú, Canaza-Cayo et al. (2024) establecieron como objetivo mejorar la precisión de la estimación del peso corporal de ovejas Corriedale sin recurrir a equipos costosos. Para ello, recopilaron datos de 100 ovinos utilizando 14 medidas corporales y aplicaron seis algoritmos de aprendizaje automático. El modelo Random Forest obtuvo el mejor desempeño con altos valores de R^2 y bajos errores (MAE, RMSE, MAPE). Este antecedente destaca el potencial del aprendizaje automático en el contexto ganadero peruano, siendo relevante para el desarrollo de modelos de predicción adaptados a condiciones locales.

En la investigación de Ormeño-Ayala & Zapata-Ttito (2024) el objetivo fue diseñar un sistema automático de estimación del peso del cuy para optimizar la producción cárnica sin generar estrés. Adaptaron la red convolucional Mask R-CNN, construyendo un dataset con 6,244 imágenes etiquetadas y desarrollaron una aplicación móvil utilizando Python, TensorFlow y Keras. El modelo logró una correlación R^2 de 80.21% y un MAPE de 12.60%. Este estudio demuestra la aplicabilidad de la segmentación de imágenes y redes neuronales en animales menores, y motiva el desarrollo de soluciones similares para el ganado vacuno en regiones como San Martín.

2.2. Fundamentos teóricos

2.2.1. Estimación del peso del ganado vacuno

2.2.1.1. Importancia de la estimación del peso

Según Peng et al. (2024), la estimación del peso corporal vivo (LBW) es muy importante para manejar bien una granja, ya que ayuda a ver cómo crecen y se desarrollan los animales. Esto mejora tanto la parte de reproducción como la forma en que se les da de comer. Además, el LBW sirve como una señal clave para saber si el ganado está sano y si está aprovechando bien el alimento. Cosas como cuánto comen, si pueden tener crías, si dan leche y cómo usan el alimento, se pueden conocer gracias a este dato. Hoy en día, hay métodos que no molestan a los animales, como tomarles fotos en 2D y usar modelos de inteligencia artificial, que han demostrado ser muy buenos para evitar el estrés que causan las balanzas normales. También, usar estas herramientas en granjas grandes ayuda a ahorrar tiempo y trabajo, ya que se puede controlar el peso de los animales sin hacerles pasar por procesos incómodos.

Como señala Hou et al. (2023), medir el peso del ganado es muy importante para saber si está sano y bien alimentado, lo que ayuda a tomar buenas decisiones sobre su

cuidado y comida. Además, hacerlo de forma no invasiva, sin tocar mucho a los animales, evita que se estresen o se lastimen, lo que mejora su bienestar, sobre todo en granjas grandes. Usar sistemas automáticos para este trabajo también ahorra tiempo y esfuerzo, y se cometen menos errores, haciendo que el trabajo sea más eficiente. Gracias a los modelos de inteligencia artificial y al uso de imágenes 3D, ahora se puede calcular el peso con más exactitud, lo que es un gran avance para mejorar tanto la producción como el cuidado de los animales.

2.2.1.2. Métodos tradicionales de estimación del peso

Según X. Li et al. (2023), los métodos tradicionales para calcular el peso del ganado usan herramientas como cintas métricas y varas para medir partes del cuerpo, como el largo del lomo, la altura del lomo, la altura de la cadera, el ancho del pecho y el abdomen. Aunque estos métodos pueden dar una idea del peso, son trabajosos y estresan a los animales, lo que puede afectar su salud y valor en el mercado. Otro método es mirar y tocar al animal para darle una nota según cómo se vea su cuerpo. Pero este sistema depende mucho de la experiencia de quien lo hace, así que puede variar mucho. Además, los dos métodos cuestan bastante y no son muy útiles en granjas grandes, porque hacen que el manejo del ganado sea más difícil y menos cómodo para los animales.

Como explica Chen et al. (2023), los métodos tradicionales para saber cuánto pesa un animal incluyen el uso de básculas, donde se pesa directamente al animal, como en el caso de los bovinos o cerdos, que deben estar en contacto con la balanza. Aunque este método es altamente preciso, requiere mover a los animales, lo que puede ser un proceso laborioso y estresante tanto para ellos como para los operarios. Por otro lado, la medición indirecta se basa en la visión computacional, que captura imágenes de los cerdos y extrae características físicas para predecir su peso. Este método evita tocar mucho al animal, lo que ayuda a que no se estrese. Sin embargo, puede fallar si el animal se mueve mucho o no está bien parado al momento de tomar la foto, ya que eso puede hacer que el cálculo del peso no sea tan exacto.

2.2.1.3. Factores que influyen en el peso del ganado

Como lo menciona Ling et al. (2022), hay varios factores que pueden hacer que la estimación del peso del ganado sea más o menos precisa. Uno de los más importantes es cómo está parado el animal al momento de medirlo. Si mueve la cabeza o tuerce el cuerpo, eso puede hacer que el cálculo del peso no sea correcto. También, los métodos manuales pueden fallar si el animal está estresado o si la persona que mide está cansada. Por otro lado, los métodos automáticos con cámaras y computadoras son más exactos y no molestan al animal. Las cámaras 3D, por ejemplo, permiten ver mejor el

tamaño y la forma del cuerpo del ganado, lo que mejora el cálculo del peso. Aun así, cosas como la luz del lugar o el fondo pueden influir en la calidad de las imágenes y afectar los resultados.

Según Temenos et al. (2024), el peso del ganado puede verse afectado por muchos factores. Por ejemplo, problemas de salud como parásitos, virus, bacterias o enfermedades que afectan su cuerpo pueden hacer que pierdan peso. También influyen cosas como la edad, si están dando leche o en qué etapa de producción se encuentran. Otro punto importante es el lugar que tienen dentro del grupo: los animales que están en un rango más bajo suelen pesar menos. Además, el clima extremo también puede afectar su peso, ya que los cambios fuertes de temperatura o un mal ambiente pueden hacer que los animales no crezcan bien. Por eso, es clave darles un espacio adecuado y saludable para que se desarrollen de forma correcta.

2.2.2. Modelos híbridos basados en segmentación de imágenes y regresión

2.2.2.1. Segmentación de imágenes

Según Ormeño-Ayala & Zapata-Ttito (2024), la segmentación de imágenes es un paso importante en el análisis visual. Consiste en marcar con precisión la parte de la imagen donde está el objeto que queremos estudiar. En este caso, se usó la plataforma Labelbox para marcar bien el área donde aparece cada cuy en las fotos. El objetivo de esta tarea es poder identificar y separar al animal del fondo, lo que hace más fácil analizarlo después. Esta parte es muy importante para entrenar modelos como Mask R-CNN, que necesitan imágenes claras y bien marcadas para poder hacer predicciones correctas, como calcular el peso del cuy.

Según Xu et al. (2024), hay varias técnicas comunes que se usan para segmentar imágenes y así poder predecir el peso del ganado. Una de las más importantes es la segmentación semántica, que usa modelos avanzados como ResNet-101-D junto con mecanismos especiales llamados SE, para identificar bien la forma del cuerpo de las vacas. También se aplican técnicas de visión por computadora que calculan distancias y tamaños del cuerpo usando modelos como EfficientNet y otros que prestan atención a detalles importantes. Además, se usan algoritmos de inteligencia artificial, como regresión lineal, máquinas SVM y redes neuronales, para predecir el peso usando medidas del cuerpo. Por último, hay técnicas llamadas de aumento de datos, que modifican las imágenes girándolas, cambiándoles el tamaño o volteándolas, para que el modelo aprenda mejor y funcione bien con distintos tipos de imágenes.

2.2.2.2. Regresión en modelos híbridos

Según lo expuesto por Z. Liu et al. (2023), los modelos híbridos combinan métodos estadísticos con técnicas de inteligencia artificial, lo que los hace útiles para resolver problemas tanto de clasificación como de predicción de valores. Uno de los más usados en este tipo de modelo son los árboles de decisión, conocidos como CART, que dividen los datos según sus características y ayudan a encontrar patrones complejos. Este tipo de modelo aprovecha lo mejor de ambos lados: la facilidad para entender los métodos estadísticos y la fuerza de los algoritmos modernos para predecir. Para saber qué tan bien funcionan estos modelos al hacer predicciones, se usan medidas como la precisión y el error cuadrático medio (RMSE), que muestran si el modelo está dando resultados confiables.

Según Ruchay et al. (2022), la regresión de imágenes es una forma eficaz de predecir el peso del ganado usando fotos normales (RGB) y mapas de profundidad. Esta técnica tiene varias ventajas, como que no hace falta usar muchos sensores al mismo tiempo, lo que ayuda a reducir costos. Se han creado tres modelos que usan estas imágenes y redes neuronales, y uno de ellos logró una precisión del 91.6% al calcular el peso. Para que el modelo funcione mejor, se usan técnicas de preparación y mejora de datos, como hacer cambios en las imágenes en 3D, lo que ayuda a tener más variedad y calidad en el entrenamiento. Este método ha demostrado ser útil en la ganadería, porque mejora la precisión de las mediciones y cuida el bienestar de los animales al evitar métodos que los puedan estresar.

2.2.2.3. Evaluación y validación de modelos híbridos

Como destaca Zhang et al. (2021), evaluar los modelos híbridos es muy importante para saber si realmente funcionan bien al calcular cosas como el peso y el tamaño del ganado. Esto ayuda a asegurarse de que los modelos sean confiables y se puedan usar en la vida real. Para probarlos, se usan datos diferentes a los que se usaron para entrenarlos, así se puede ver si el modelo es capaz de dar buenos resultados en casos nuevos. Para medir qué tan bien funciona, se usan medidas como el error medio (MAE), el error cuadrado medio (MSE) y el valor R^2 , que muestran si el modelo está haciendo buenos cálculos. Además, se comparan varios modelos como DenseNet201, ResNet152 V2, Xception y MobileNet V2, para ver cuál da los mejores resultados usando los mismos datos de prueba.

Como explica J. Liu et al. (2021), evaluar y comprobar que los modelos híbridos funcionan bien es clave para asegurar que puedan predecir y clasificar con precisión. Para esto, se usan diferentes medidas, como las curvas ROC, el índice SCAI y la

precisión en la clasificación, que ayudan a ver el rendimiento general del modelo. Se ha visto en varios estudios que los modelos híbridos, como el CNN-FMV, funcionan mejor que los modelos usados por separado. Comparando, combinaciones como SVM-FMV, CART-FMV y CNN-FMV dan mejores resultados que los algoritmos cuando se usan solos. En casos específicos, como ver qué zonas pueden inundarse, estos modelos han demostrado que entre el 13.21% y el 22.03% de las áreas tienen riesgo alto o muy alto, lo que muestra que los modelos híbridos son muy útiles para detectar riesgos y tomar decisiones basadas en datos reales.

2.2.2.4. Conjunto de datos de imágenes

Un conjunto de datos de imágenes es un grupo de fotos organizadas con información extra sobre cada una, y se usa para un propósito específico, como hacer investigaciones. En el caso de la visión por computadora y la inteligencia artificial, estos conjuntos sirven para entrenar y probar modelos, ya que incluyen tanto las imágenes como datos que explican lo que aparece en ellas. Son muy importantes en estudios como los de biodiversidad, donde ayudan a analizar cómo se ven y se comportan distintas especies (R. Li et al., 2024; Löffler et al., 2021; Los et al., 2023) .

2.2.2.5. Coeficiente de determinación (R^2)

Es una medida que muestra qué tan bien los resultados reales se ajustan a lo que predice un modelo de regresión. Si el valor de R^2 está cerca de 1, quiere decir que el modelo explica casi toda la variación de los datos, lo que significa que hace predicciones muy precisas (J. Liu et al., 2023; J. Zhang et al., 2021).

2.2.2.6. Error absoluto medio (MAE)

El Error Absoluto Medio (MAE) es una forma de medir qué tan bien está funcionando un modelo. Se calcula sacando el promedio de las diferencias, en valor absoluto, entre lo que el modelo predice y los valores reales. No importa si el error es hacia arriba o hacia abajo, solo cuánto se equivocó. Si el MAE es bajo, quiere decir que el modelo predice con más precisión y se acerca más a los resultados reales (Canaza-Cayo et al., 2024; da Cunha et al., 2024).

2.2.2.7. Raíz del error cuadrático medio (RMSE)

La Raíz del Error Cuadrático Medio (RMSE) es una medida que sirve para saber qué tan lejos están las predicciones de un modelo con respecto a los valores reales. Se calcula sacando la raíz cuadrada del promedio de los errores al cuadrado. Si el RMSE es bajo, eso significa que el modelo se está equivocando poco y sus predicciones están muy cerca de los datos reales, lo que indica que funciona con buena precisión (Deng et al., 2025; J. Liu et al., 2021).

2.2.2.8. Red Neuronal Convolutacional (CNN)

La Red Neuronal Convolutacional (CNN) es un tipo especial de red que se usa principalmente para trabajar con imágenes. A diferencia de otras redes, las CNN usan una operación llamada 'convolución', que les permite identificar patrones y detalles dentro de las imágenes. Gracias a esto, pueden entender cómo se relacionan los píxeles entre sí y reconocer formas, bordes y otras características. Esto las hace muy buenas para tareas como clasificar imágenes o separar objetos dentro de una foto (Hou et al., 2025; J. Li et al., 2023; Temenos et al., 2024).

2.2.2.9. Visión por computadora

La Visión por Computadora es una tecnología que usa cámaras y programas especiales para observar y analizar imágenes o videos de animales en las granjas. Con estos sistemas, se puede obtener información útil sin molestar a los animales y con un costo bajo. Sus componentes principales incluyen cámaras, unidades de grabación y procesamiento, así como modelos que analizan la información para determinar la calidad de los resultados (Bi et al., 2023; G. Li et al., 2022).

2.2.2.10. YOLOv8

YOLOv8 es la última versión mejorada del modelo de detección YOLO, utilizado para la detección en tiempo real de objetos, como la postura e identificación de yaks. Incluye cuatro componentes: el Input, que mejora la diversidad de los datos; el Backbone, que utiliza una versión eficiente de Darknet53 para la extracción de características; el Neck, que mejora la precisión con una Red de Pirámide de Características; y el Head, que maneja tareas de clasificación y regresión de forma independiente, optimizando la precisión y velocidad del modelo (Peng et al., 2024).

CAPÍTULO III MATERIALES Y MÉTODOS

3.1. **Ámbito y condiciones de la investigación**

3.1.1. **Contexto de la investigación**

La presente investigación se desarrolló en el distrito de Zapatero, ubicado en la provincia de Lamas, región San Martín, al noreste del Perú. Este territorio pertenece a la selva alta y presenta una geografía diversa compuesta por colinas, valles y zonas de bosque húmedo tropical, condiciones que han favorecido históricamente la actividad agropecuaria, especialmente la ganadería de bovinos. En este distrito se encuentra el fundo “Ganadería DR”, dedicado a la cría y manejo de ganado vacuno, contribuyendo significativamente a la actividad agropecuaria local (Figura 1).

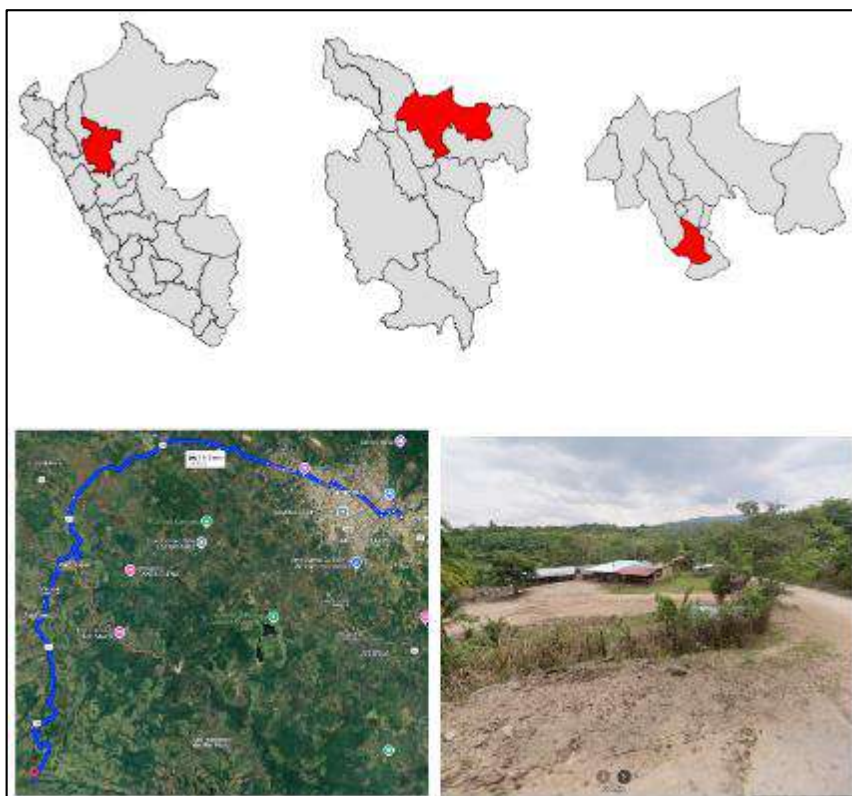


Figura 1
Ubicación de la investigación

3.1.2. **Periodo de ejecución**

La investigación se ejecutó entre enero y julio del año 2025, conforme al cronograma de actividades establecido en el proyecto. Durante este periodo se desarrollaron las fases de planificación, captura de imágenes, registro del peso real del ganado, construcción del conjunto de datos, entrenamiento de modelos híbridos y evaluación de su

desempeño. Se hicieron todas las actividades a tiempo, lo que ayudó a cumplir los objetivos paso a paso y de forma ordenada.

3.1.3. Autorizaciones y permisos

Para la ejecución del presente estudio, se gestionó la autorización necesaria mediante un documento en el cual el propietario del predio otorgó su consentimiento para la realización de las actividades de recolección de datos. Este permiso permitió el uso de las instalaciones para el registro de imágenes y el pesaje del ganado vacuno. La autorización fue proporcionada directamente por el dueño del fundo, quien expresó su conformidad para que la investigación se desarrollara en su propiedad, asegurando el acceso adecuado y las condiciones logísticas requeridas durante todo el proceso (Anexo 01).

3.1.4. Control ambiental y protocolos de bioseguridad

Durante el estudio, se siguieron todas las reglas de cuidado del medio ambiente y las medidas de seguridad para trabajar con animales al aire libre. Las actividades se hicieron en lugares con buena ventilación y sin hacer nada que cambiara el comportamiento normal del ganado. El equipo estuvo formado por personas con experiencia en el manejo de vacas, lo que ayudó a tratarlas bien y con responsabilidad. También se usaron normas de limpieza, se mantuvo distancia con los animales y todos los del equipo usaron ropa y equipos especiales para protegerse. Gracias a todo esto, se cuidó tanto a los animales como al personal, siempre pensando en el bienestar de todos.

3.1.5. Aplicación de principios éticos internacionales

La investigación se hizo respetando las reglas éticas que se aplican cuando se trabaja con seres vivos. En todo momento se actuó con honestidad, cuidando que los datos sean reales y el método usado sea claro. A las personas que estaban a cargo de los animales se les pidió permiso para tomar las fotos, y se cuidó que su información sea privada. También se protegió el bienestar de los animales, sin hacerles daño ni causarles estrés, siguiendo siempre el principio de hacer el bien. Asimismo, se respetó el ecosistema natural donde se realizaron las actividades, y se actuó bajo criterios de equidad y justicia, sin generar ningún perjuicio a los participantes ni al entorno.

3.2. Sistema de variables

3.2.1. Variables principales

Variable dependiente: Pesaje no invasivo de ganado vacuno.

Variable independiente: Modelos híbridos basados en segmentación de imágenes y regresión.

Tabla 1

Descripción de variables por objetivo específico 1

Variable abstracta	Variable concreta	Medio de registro	U. de med.
Pesaje no invasivo de ganado vacuno.	Fotografías de las características morfológicas del ganado vacuno.	Cámara digital de alta resolución.	Unidad.
	Peso registrado del ganado.	Balanza digital.	Kilogramos (kg).

Tabla 2

Descripción de variables por objetivo específico 2

Variable abstracta	Variable concreta	Medio de registro	U. de med.
Modelos híbridos basados en segmentación de imágenes y regresión.	Error absoluto medio (MAE)	Índice estadístico	Porcentaje
	Raíz del Error cuadrático medio (RMSE)	Índice estadístico	Porcentaje
	Coef. de determinación (R^2)	Índice estadístico	Porcentaje

3.2.2. Variables secundarias

No aplica.

3.3. Procedimientos de la investigación

La investigación fue de tipo aplicada, ya que tuvo como propósito brindar una solución tecnológica práctica al problema del pesaje del ganado vacuno mediante un enfoque no invasivo, utilizando modelos híbridos que combinaron técnicas de visión artificial con algoritmos de regresión. Asimismo, el estudio se enmarcó en un nivel experimental, dado que contempló la manipulación controlada de variables mediante la evaluación comparativa de diversos modelos predictivos basados en datos reales recolectados en condiciones de campo. Para ello, se empleó un diseño factorial de tipo 3x1, el cual permitió analizar el efecto de tres modelos de regresión distintos sobre una variable común, facilitando así la comparación sistemática de su rendimiento.

La población del estudio estuvo conformada por el conjunto total de imágenes digitales de ganado vacuno obtenidas durante el proceso de recolección de datos en el fundo "Ganadería DR", perteneciente a un poblador del distrito de Zapatero, ubicado en la provincia de Lamas, región San Martín. Estas imágenes capturaron perfiles laterales de bovinos que se encontraban en condiciones adecuadas de salud y movilidad, permitiendo asegurar la calidad de los datos obtenidos.

A partir de esta población, se obtuvo una muestra representativa compuesta por aproximadamente 3500 imágenes, correspondientes a múltiples ejemplares provenientes del mismo fundo y de otras fincas de la zona. Esta diversidad morfológica

presente en las imágenes garantizó la variabilidad necesaria para entrenar, validar y comparar el rendimiento de los modelos híbridos desarrollados en el estudio.

El diseño muestral adoptado fue no probabilístico por conveniencia, basado en la disponibilidad de ganado vacuno durante las jornadas de pesaje, realizadas entre las 7:00 a.m. y las 12:00 p.m. Las imágenes de cada ejemplar fueron tomadas y asociadas a su peso real para ser utilizadas en el entrenamiento de los modelos.

El diseño experimental fue factorial 3x1, evaluando tres modelos híbridos que combinaron YOLOv8 para la segmentación de las imágenes con algoritmos de regresión: XGBoost, Random Forest y LightGBM. Los modelos fueron entrenados con el mismo conjunto de características extraídas para comparar su rendimiento en la predicción del peso del ganado vacuno.

El diseño analítico consistió en evaluar el rendimiento de los modelos mediante las métricas R^2 , RMSE y MAE. R^2 permitió evaluar la capacidad del modelo para explicar la variabilidad del peso del ganado vacuno, mientras que RMSE y MAE proporcionaron una medición del error promedio en las predicciones, permitiendo determinar cuál modelo producía las estimaciones más precisas.

En cuanto a la representación de la información, se organizó una base de datos estructurada que vinculó cada imagen con su peso correspondiente. Se aplicaron procesos de limpieza y validación, eliminando registros con imágenes borrosas, deformadas o etiquetadas incorrectamente, a fin de asegurar la consistencia del conjunto de datos y la fiabilidad de los resultados.

Para el análisis estadístico, se calcularon métricas estándar de evaluación de modelos de regresión, como el Error Absoluto Medio (MAE), la Raíz del Error Cuadrático Medio (RMSE) y el Coeficiente de Determinación (R^2). Estas métricas permitieron medir el grado de precisión de las predicciones en relación al peso real registrado del ganado.

No obstante, para efectos comparativos entre modelos y la realización del análisis de varianza (ANOVA), se eligió el RMSE como métrica principal, debido a su capacidad para penalizar con mayor severidad los errores grandes, aspecto crítico en la estimación del peso, donde desviaciones significativas podrían implicar pérdidas económicas. Además, el RMSE es ampliamente utilizado en la literatura especializada como referencia estándar para comparar modelos de regresión en el ámbito ganadero y agrícola (Wang et al., 2025; Xu et al., 2024).

3.3.1. Actividades del objetivo específico 1: Construir un conjunto de datos mediante la toma de imágenes de ganado vacuno asociadas a su peso real en un centro de pesaje.

Con el objetivo de desarrollar un modelo computacional que estime el peso del ganado vacuno a partir de imágenes, fue necesario construir un conjunto de datos robusto, trazable y representativo. Este conjunto debía incluir fotografías de perfil de los ejemplares, capturadas bajo condiciones controladas, y su correspondiente peso corporal real, obtenido mediante balanza electrónica de precisión.

Para asegurar la calidad y consistencia de los datos, se implementó un protocolo sistemático dividido en cinco etapas consecutivas. Estas fases abordaron desde la adecuación del entorno físico hasta la asociación final de los registros, garantizando la trazabilidad de cada ejemplar en todas las fases del proceso. A continuación, se presenta un diagrama explicativo (Figura 2).



Figura 2

Diagrama de etapas del objetivo específico 1

Las actividades del objetivo se explican de la siguiente manera:

Primera etapa: Adecuación del entorno

La primera acción consistió en identificar e intervenir un espacio físico adecuado para la ejecución del procedimiento de captura y registro de datos (**Figura 3**). Para ello, se utilizó el fundo “Ganadería DR”, ubicado en el distrito de Zapatero, provincia de Lamas, el cual fue acondicionado para cumplir con los requerimientos del estudio. Se construyó una pasarela que conectó el área de ingreso con la balanza, permitiendo conducir el

ganado de manera segura y controlada. Esta infraestructura fue diseñada para minimizar el estrés de los animales y facilitar su posicionamiento para el registro de su peso real, reduciendo al mismo tiempo posibles interferencias externas durante el proceso (Brando & Norman, 2023; Ogawa et al., 2024).

La balanza utilizada fue una plataforma de pesaje tipo jaula, con dimensiones de 270 cm de largo, 100 cm de ancho y 200 cm de alto, especialmente diseñada para trabajos continuos y pesados. Cuenta con una estructura reforzada en ángulo $\frac{3}{4}$, planchas estriadas de acero de 3/16 pulgadas de espesor, puertas corredizas de entrada y salida, y un indicador electrónico de alta precisión fabricado en polímero ABS resistente, con pantalla LCD retroiluminada. Además, incorporaba funciones de tara, retención de peso (hold) para animales en movimiento, y apagado automático para prolongar la vida útil de la batería. El sistema opera con alimentación de corriente alterna (110V/220V) o batería interna recargable, permitiendo una autonomía de hasta 96 horas. Estas características aseguraron precisión, durabilidad y fiabilidad durante todo el proceso de recolección de datos (Hou et al., 2023; H. Zhang et al., 2025).



Figura 3
Adecuación del entorno físico para la recolección de datos

Segunda etapa: Registro del peso

Después de adecuar el entorno, cada animal fue conducido por la pasarela hacia la balanza electrónica instalada en el centro de pesaje, equipada con cuatro sensores Mavin, cada uno con capacidad de tres toneladas. Esta balanza permitió registrar de forma precisa y digital el peso corporal del ejemplar en kilogramos (Figura 4). La precisión en esta etapa es fundamental para garantizar la calidad del conjunto de datos (Canaza-Cayo et al., 2024; Paudel et al., 2023).

El peso obtenido fue asociado inmediatamente a un código numérico único asignado al ejemplar, el cual se utilizaría posteriormente en la etapa de marcaje visual. Esta asociación garantizó una correcta correspondencia entre los registros de peso y las imágenes capturadas posteriormente, asegurando la integridad y trazabilidad del conjunto de datos (Firdaus et al., 2024).



Figura 4
Registro del peso real del ganado

Tercera etapa: Colocación del objeto de referencia

Después de registrar el peso del ejemplar en la balanza, se procedió a preparar al animal para la captura de imágenes. Como parte de este proceso, se utilizó un objeto de referencia físico: un círculo de vinil de seis pulgadas de diámetro, fabricado con material liviano, flexible y autoadhesivo, ideal para fijarse al cuerpo del animal sin causar incomodidad ni afectar su movilidad (Figura 5).

Sobre dicho círculo se escribió manualmente un código numérico único, previamente asignado al ejemplar. Este código permitió identificar visualmente al animal en las imágenes y garantizar la correcta vinculación con su peso registrado, asegurando la trazabilidad individual en el conjunto de datos y evitando errores de asociación durante las fases posteriores del análisis.

Si bien el código escrito cumplió la función de identificación, el círculo en sí actuó como objeto de referencia visual, permitiendo estandarizar la escala en todas las imágenes. Su presencia en el plano lateral del cuerpo del animal facilitó el ajuste de distancia y tamaño durante el preprocesamiento de imágenes, contribuyendo a que las características morfométricas extraídas fueran comparables entre ejemplares.

El uso de marcadores visuales para referencia es una práctica frecuente en sistemas de visión por computadora para ganado, ya que permite estandarizar escalas y mejorar la precisión de las mediciones (X. Li et al., 2023; Ling et al., 2022).



Figura 5
Colocación del objeto de referencia

Cuarta etapa: Captura de imágenes de perfil

Con el objeto de referencia ya colocado, se procedió a la captura fotográfica del perfil del animal. Para esta tarea se utilizó un teléfono móvil Samsung Galaxy A52s 5G, equipado con una cámara trasera principal de 64 MP con apertura f/1.8, enfoque automático y estabilización óptica de imagen. A diferencia de sistemas de captura con distancia o ángulo fijo, en este estudio se optó por una estrategia flexible, aprovechando momentos en los que el animal permanecía quieto y en posición de perfil. Esta decisión respondió a condiciones prácticas del entorno y al comportamiento del ganado,

asegurando imágenes útiles sin interferir con su bienestar (Hou et al., 2023; Kumar et al., 2024).

El encuadre se ajustó de manera que la imagen incluyera el cuerpo completo del animal, asegurando una vista lateral clara y perpendicular al plano de la cámara (Figura 6). Se prestó especial atención a que el animal permaneciera estático durante la toma, reduciendo al mínimo las posibles distorsiones causadas por el movimiento o por ángulos inadecuados (Z. Liu et al., 2023).



Figura 6
Captura de imágenes del perfil del ganado vacuno

Quinta etapa: Asociación imagen-peso

Una vez finalizada la captura de imágenes, se realizó el proceso de organización y vinculación entre los datos fotográficos y los registros de peso. Para ello, cada imagen fue revisada y renombrada en función del código numérico visible en el círculo adherido al animal. Este código permitió establecer una asociación directa entre la imagen y el valor de peso registrado previamente, lo cual fue esencial para conformar un conjunto de datos supervisado confiable (Afridi et al., 2024; Jiang et al., 2024).

La estructura final resultante consistió en una colección de imágenes etiquetadas, donde cada archivo estaba vinculado a un valor real de peso corporal (**Figura 7**). Durante esta

etapa se aplicaron criterios de organización, lo que facilitó la gestión del repositorio de datos y aseguró la trazabilidad completa de cada ejemplar dentro del estudio. La correcta asociación entre imágenes y pesos fue clave para evitar errores en la etapa de modelado y garantizar la validez de los resultados obtenidos (Ma et al., 2024).

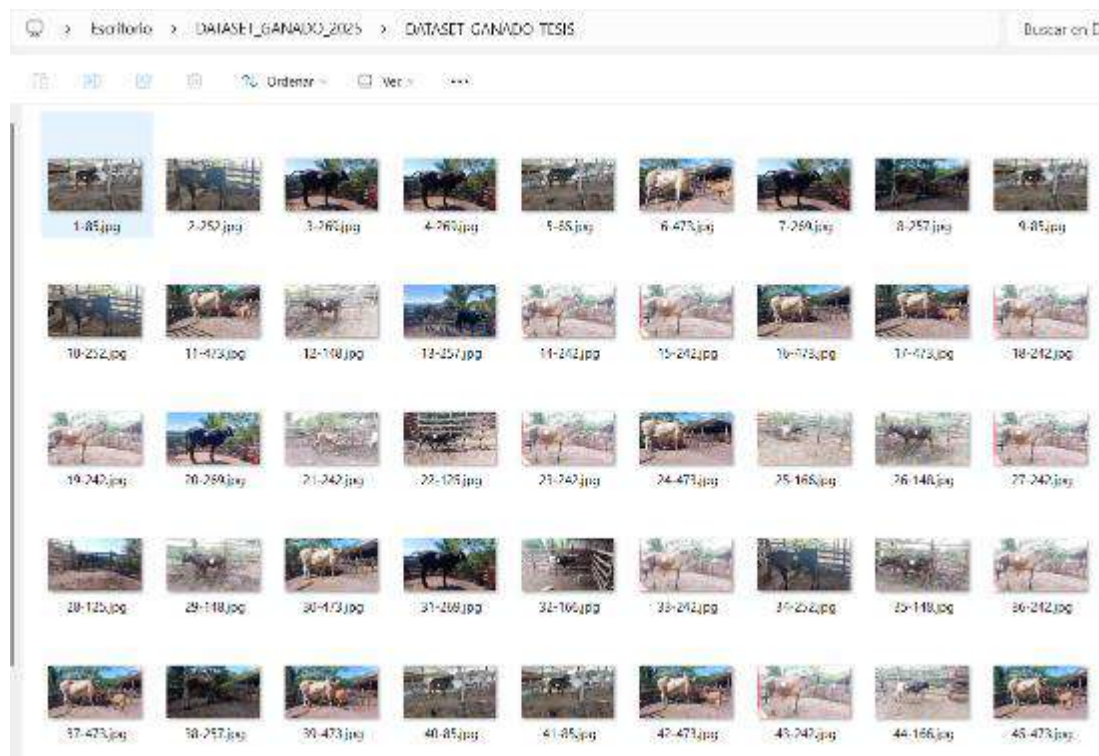


Figura 7
Imágenes nombradas con código y peso real de gana ejemplar.

3.3.2. Actividades del objetivo específico 2: Desarrollar modelos híbridos basados en segmentación de imágenes y regresión para estimar de forma no invasiva el peso del ganado vacuno.

Con el objetivo de estimar el peso del ganado vacuno de forma no invasiva, se desarrolló modelos híbridos basado en segmentación de imágenes y regresión. El proceso se estructura en dos fases principales: primero, se realiza la carga de imágenes del ganado, donde cada archivo contiene como metadato el peso real del ejemplar. Luego, se detecta un círculo de referencia para estandarizar proporciones y se segmenta automáticamente la silueta del animal utilizando el modelo YOLOv8 preentrenado.

En la segunda fase, se extraen características geométricas relevantes como área, perímetro, altura, ancho, relación de aspecto y momentos de Hu. Estas características se vinculan con el peso real de cada ejemplar, permitiendo entrenar modelos de regresión validados con KFold ($k=5$) y optimizados mediante RandomizedSearchCV. A continuación, se presenta un diagrama explicativo del proceso seguido (Figura 8).

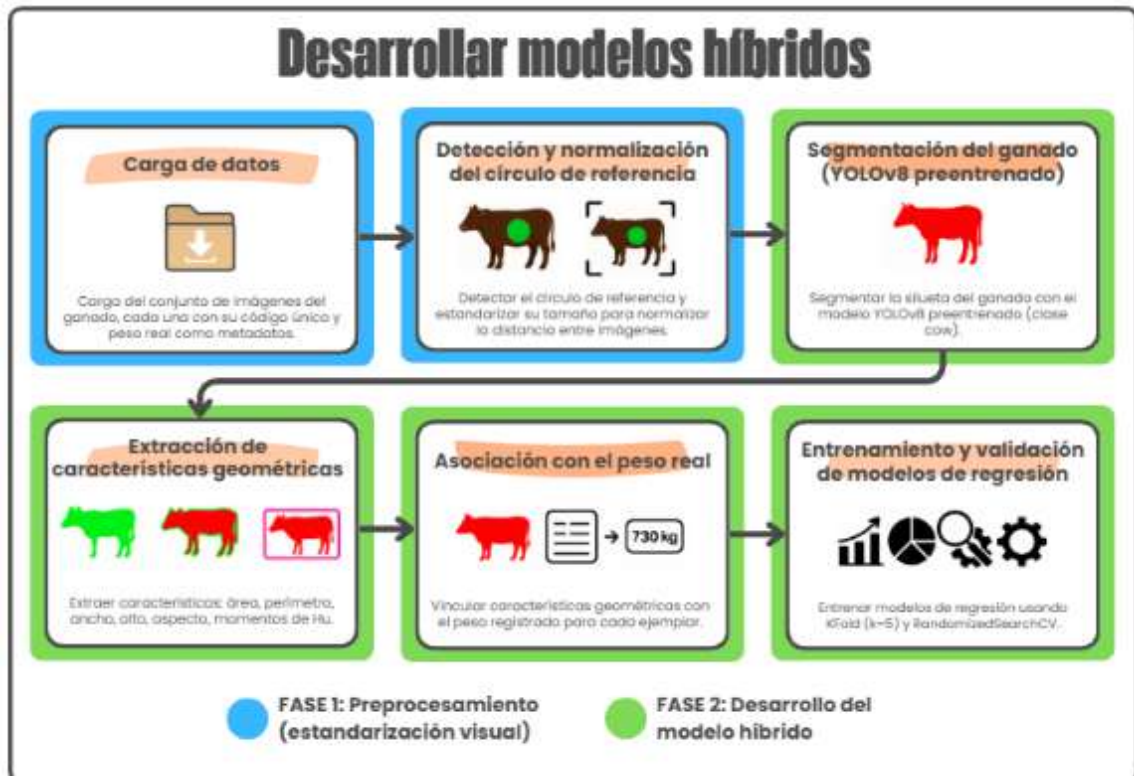


Figura 8
Diagrama de etapas del objetivo específico 2

FASE 1: Preprocesamiento (estandarización visual)

Etapa 1: Carga de datos

En esta primera etapa, se procedió a la carga del conjunto de imágenes correspondiente al ganado vacuno, previamente almacenado en una carpeta específica dentro del entorno de Google Drive. Cada imagen fue nombrada siguiendo una convención estructurada, donde el nombre del archivo incluye un identificador único y el peso real del ejemplar (por ejemplo, 1-85.jpg, donde 85 representa el peso en kilogramos). Esta estrategia permitió vincular de manera directa la imagen con su valor de referencia, facilitando así su uso como metadato durante las fases de entrenamiento del modelo. Tal enfoque resulta clave en aplicaciones actuales de visión computacional aplicadas a estimaciones biométricas en el ámbito ganadero (Firdaus et al., 2024; Ma et al., 2024).

Para habilitar el acceso al conjunto de datos desde el entorno de desarrollo, se empleó la plataforma Google Colab, aprovechando su funcionalidad para montar unidades de almacenamiento en la nube. Tras realizar la autenticación del usuario, se accedió a la carpeta correspondiente dentro del sistema de archivos, desde donde fue posible visualizar y listar el contenido del dataset.

Posteriormente, se estableció un procedimiento programático que permitió identificar cada archivo de imagen, extraer de su nombre el valor de peso real y organizar estos

datos en estructuras que facilitaron su posterior uso en los procesos de segmentación y regresión. Este método no solo garantizó la trazabilidad de los datos, sino también una adecuada estructuración del conjunto de imágenes, de acuerdo con las buenas prácticas recomendadas en la literatura para la gestión de datasets agropecuarios (Arshad et al., 2024; Löffler et al., 2021) (Figura 9).

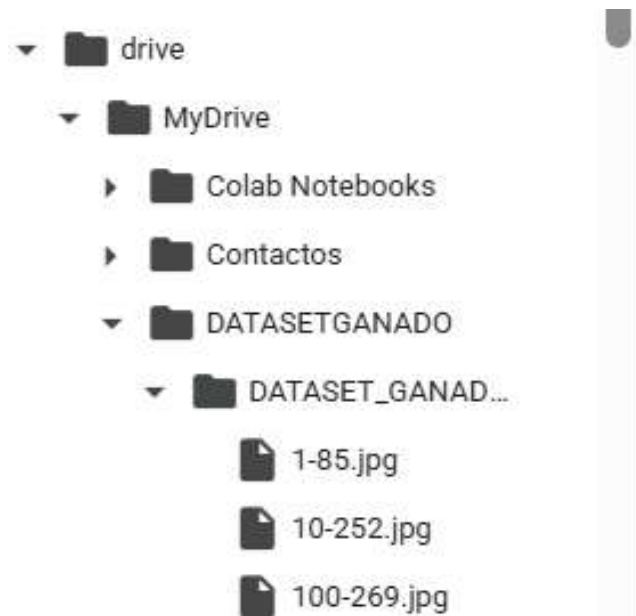


Figura 9.
Estructura del dataset en Google Drive

Etapa 2: Detección y normalización del círculo de referencia

Con el objetivo de estandarizar la escala visual de todas las imágenes del ganado vacuno, se implementó una fase de preprocesamiento centrada en la detección de un objeto de referencia: un círculo físico de seis pulgadas de diámetro, adherido lateralmente al cuerpo de cada animal. La presencia de este marcador permitió simular una calibración de distancia en condiciones de campo, alineándose con metodologías previamente aplicadas en trabajos de medición animal con visión computacional (Hou et al., 2023; Peng et al., 2024).

Para detectar el círculo de referencia, se usó un modelo personalizado basado en la arquitectura YOLOv8, que fue entrenado específicamente para reconocer esa clase. El modelo analizó una por una las imágenes, y las que estaban en posición vertical se giraron automáticamente a horizontal, para que todas tuvieran la misma orientación y se mantuviera la coherencia en el conjunto.

Una vez que se detectó el círculo, se recortó su zona usando las máscaras que generó el modelo YOLOv8. Con esa parte segmentada, se midió el radio del círculo en píxeles

y, a partir de eso, se calculó un factor de escala. Este factor sirvió para ajustar el tamaño de toda la imagen, de modo que el círculo tuviera el tamaño real de seis pulgadas. Así, se pudo escalar todas las imágenes por igual y mantener una medida uniforme. Esta práctica es ampliamente recomendada para minimizar distorsiones geométricas y asegurar la comparabilidad de características morfológicas entre individuos (X. Li et al., 2023; H. Zhang et al., 2025).

Cuando la imagen ajustada no tenía el tamaño necesario para los siguientes pasos, se le agregó un borde blanco (padding) alrededor, centrando el contenido. Esto ayudó a que todas las imágenes tuvieran el mismo aspecto y tamaño, manteniendo la uniformidad entre todas las muestras.

Finalmente, las imágenes preprocesadas fueron convertidas al formato RGB y almacenadas en una carpeta destinada exclusivamente al dataset normalizado. Este procedimiento aseguró que todas las muestras compartieran una referencia geométrica común, indispensable para que las características extraídas fueran consistentes, independientemente de la distancia o el ángulo desde el cual se capturó la imagen (Kumar et al., 2024; X. Li et al., 2023; Xiong et al., 2023).

FASE 2: Desarrollo de modelos híbridos

Etapa 3: Segmentación del ganado (YOLOv8 preentrenado)

Una vez estandarizadas las imágenes mediante la referencia circular, se procedió a la segmentación automática del ganado utilizando un modelo preentrenado de YOLOv8 en su variante para segmentación (yolov8x-seg.pt). Este modelo, entrenado previamente con la clase cow, permitió identificar y aislar la silueta del animal sin requerir entrenamiento adicional, constituyendo así una solución eficiente para la clasificación morfológica en escenarios ganaderos (Devi et al., 2024; Peng et al., 2024).

El proceso de inferencia fue aplicado de forma automatizada sobre cada imagen del conjunto preprocesado. A partir de la predicción generada por el modelo, se obtuvieron las coordenadas asociadas a la máscara de segmentación correspondiente al cuerpo del animal. Estas coordenadas fueron convertidas en polígonos y procesadas mediante técnicas de visión computacional para generar una máscara binaria: una imagen en la que la silueta de la vaca se representó en color rojo, mientras que el fondo fue sustituido por blanco, permitiendo así una mayor claridad en la lectura visual y computacional.

Posteriormente, cada imagen segmentada fue redimensionada a un tamaño fijo de 256 × 256 píxeles, garantizando la uniformidad del dataset y facilitando su integración en modelos de aprendizaje profundo. Las imágenes resultantes fueron almacenadas en un

directorio especialmente diseñado para contener únicamente las muestras segmentadas, completamente preparadas para la siguiente fase de extracción de características.

Este procedimiento automatizado permitió la generación de un conjunto de datos limpio, estructurado y morfológicamente consistente, donde cada ejemplar bovino aparece claramente delimitado. Esta estrategia se encuentra alineada con las buenas prácticas actuales en visión computacional agropecuaria, en las cuales la segmentación precisa constituye una etapa crítica para el análisis cuantitativo posterior (Afridi et al., 2024; Firdaus et al., 2024).

Etapa 4: Extracción de características geométricas

Después de la segmentación, se pasó a medir las formas del cuerpo de cada animal usando las imágenes ya procesadas. En esta etapa se sacaron características geométricas, es decir, datos sobre la forma y estructura del cuerpo. Esto fue clave para poder estimar el peso con los modelos de regresión. El procedimiento se basó en técnicas ampliamente aplicadas en estudios de evaluación corporal animal a partir de imágenes (Brando & Norman, 2023; G. Li et al., 2021).

Primero, cada imagen se convirtió a blanco y negro (escala de grises) para que fuera más fácil trabajar con ella. Luego, se aplicó un filtro que separó bien la figura del animal del fondo blanco, permitiendo identificar sus bordes con precisión. Después, se analizaron los contornos en cada imagen y se eligió el más grande como la silueta principal del animal. A partir de esta silueta, se calcularon medidas importantes, como el área en píxeles, el largo del borde (perímetro), el ancho y el alto máximos, y la relación entre ellos (relación de aspecto). También se calcularon tres Momentos de Hu, que ayudan a describir la forma del animal y no cambian, aunque la imagen se gire, se mueva o se haga más grande o más chica.

La elección de estas variables responde a su probada utilidad para reflejar diferencias estructurales vinculadas al peso corporal, incluso en condiciones no controladas o en tomas realizadas al aire libre. Investigaciones recientes han respaldado el valor predictivo del área proyectada, la relación de aspecto y los momentos de Hu como insumos confiables para modelos de regresión en ganado bovino y otras especies animales (Canaza-Cayo et al., 2024; Ling et al., 2022; Xu et al., 2024).

Este conjunto de características fue sistemáticamente extraído de cada imagen y almacenado en un formato estructurado, listo para ser integrado en la etapa final de modelado predictivo. Gracias a este procedimiento, se obtuvo una representación

matemática robusta de la morfología bovina, lo que permitió reducir la complejidad visual de las imágenes originales sin perder la información esencial para estimar el peso de forma automática y precisa.

Etapas 5: Asociación con el peso real

Después de obtener las medidas geométricas de cada imagen segmentada, se les asignó el valor real de peso correspondiente. De esta forma, se creó un conjunto de datos supervisado, que sirvió como base para entrenar los modelos de regresión en las siguientes etapas del estudio. La construcción precisa de esta estructura fue esencial para mantener la trazabilidad entre la representación visual del animal y su peso objetivo, tal como lo destacan estudios recientes sobre predicción de peso en ganado mediante técnicas de aprendizaje automático (Anjar Setiawan et al., 2024; Chen et al., 2023).

Cada grupo de datos numéricos sacados de las imágenes fue guardado como variable independiente, mientras que los pesos reales se registraron por separado como la variable dependiente. Ambos conjuntos se organizaron con el mismo orden, asegurando que cada grupo de datos estuviera correctamente ligado a su peso. Esta relación directa entre las entradas y salidas del modelo fue clave para que el aprendizaje supervisado funcionara de forma correcta y confiable.

Para que los datos pudieran ser usados fácilmente en los siguientes pasos, se organizaron en matrices especiales que los algoritmos de aprendizaje automático pueden entender sin problemas. Esta forma de organizar la información no solo permitió entrenar los modelos híbridos, sino también aplicar validación cruzada para comprobar qué tan bien funcionaban.

La correcta asociación entre los vectores de características morfológicas y los valores reales de peso aseguró que los modelos pudieran aprender de manera precisa la relación existente entre la forma del animal y su masa corporal. Esta precisión en la etapa de emparejamiento resultó crucial para sostener la confiabilidad del sistema predictivo, especialmente en aplicaciones agropecuarias donde la robustez del modelo es determinante para su utilidad práctica (Afridi et al., 2024; Paudel et al., 2023).

Etapas 6: Entrenamiento y validación de modelos de regresión

Con el conjunto de datos supervisado debidamente estructurado, se procedió al entrenamiento de modelos de regresión con el propósito de estimar el peso del ganado vacuno a partir de sus características morfológicas. Esta fase constituyó el núcleo del enfoque híbrido propuesto, al integrar técnicas tradicionales de aprendizaje automático

con información cuantificada desde imágenes, aplicando principios consolidados en estudios recientes de visión computacional aplicada al ámbito agropecuario (Firdaus et al., 2024; G. Li et al., 2022).

Para asegurar que los resultados fueran confiables, se usó la técnica de validación cruzada llamada K-Fold, dividiendo los datos en cinco partes al azar. Con este método, cada muestra se usó tanto para entrenar como para evaluar el modelo. Esto ayudó a que el modelo aprendiera de forma más general y no se ajustara demasiado a los datos, evitando así errores por sobreajuste.

En esta etapa se entrenaron varios algoritmos de regresión. Para mejorar su rendimiento, se ajustaron sus configuraciones internas usando una búsqueda aleatoria, que permitió probar diferentes combinaciones sin gastar tanto tiempo ni recursos como lo haría una búsqueda completa. Entre los parámetros que se evaluaron estaban el número de árboles, la profundidad máxima, el tamaño mínimo para dividir o crear hojas, el tipo de muestreo y el criterio para elegir las características en cada división.

Una vez finalizado el proceso de validación cruzada y ajuste, se identificó el modelo con la configuración óptima, el cual fue retenido como el estimador final. Este modelo fue posteriormente almacenado en disco con el propósito de facilitar su reutilización en la siguiente etapa de evaluación, cumpliendo así con las buenas prácticas de reproducibilidad científica en proyectos de machine learning orientados al agro (Afridi et al., 2024; Brando & Norman, 2023; Ma et al., 2024).

De esta manera, se obtuvo un modelo entrenado, validado y conservado, listo para ser sometido a una evaluación exhaustiva de su desempeño predictivo, lo que permitirá valorar su aplicabilidad práctica en la estimación automatizada del peso del ganado.

3.3.3. Actividades del objetivo específico 3: Evaluar las métricas de rendimiento de los modelos híbridos basados en segmentación de imágenes y regresión en la estimación del peso del ganado vacuno.

Con el objetivo de evaluar el rendimiento predictivo de los modelos híbridos construidos, se estableció un proceso metodológico estructurado en tres etapas principales (Figura 10). En la primera etapa se consolidaron los resultados obtenidos en las distintas ejecuciones de los modelos, organizando las métricas de desempeño en una tabla comparativa que facilitó su análisis posterior. En la segunda etapa se aplicaron técnicas estadísticas para determinar si existían diferencias significativas entre los modelos, utilizando pruebas como el ANOVA de una vía y, en caso necesario, pruebas post hoc. Finalmente, en la tercera etapa se interpretaron los resultados obtenidos y se seleccionó el modelo con mejor rendimiento, considerando tanto los promedios alcanzados como

la significancia estadística de las diferencias observadas. Este enfoque secuencial permitió analizar de manera rigurosa y objetiva el comportamiento de los modelos propuestos, asegurando una elección fundamentada del algoritmo más eficiente para la estimación automatizada del peso del ganado vacuno.



Figura 10
Diagrama de etapas del objetivo específico 3

Primera etapa: Consolidación de resultados en tabla comparativa

Una vez culminado el proceso de desarrollo de los modelos que combinan el algoritmo YOLOv8 con Random Forest Regressor, XGBoost Regressor y LightGBM Regressor, se procedió a consolidar sus métricas de rendimiento. Para ello, se organizó la información generada a partir de las dieciséis corridas realizadas, en las cuales se excluyó de manera sistemática un subconjunto diferente del conjunto de prueba con el fin de introducir aleatoriedad y aumentar la robustez del análisis. Cada corrida implicó la evaluación de los modelos sobre una muestra ligeramente distinta, evitando así sesgos y favoreciendo una estimación más generalizable de su desempeño.

Los resultados obtenidos fueron registrados en una hoja de cálculo estructurada, en la que cada fila representaba una corrida independiente y las columnas agrupaban las métricas obtenidas por cada modelo: Error Absoluto Medio (MAE), Raíz del Error Cuadrático Medio (RMSE) y Coeficiente de Determinación (R^2). Este formato tabular permitió visualizar y comparar de forma clara el comportamiento de los algoritmos en distintos escenarios de prueba, sentando las bases para el posterior análisis estadístico que permitiría sustentar con evidencia la elección del modelo más eficiente.

La estructuración de esta matriz comparativa respondió a criterios de trazabilidad y transparencia en la evaluación del rendimiento algorítmico, elementos considerados fundamentales en estudios similares donde se requiere contrastar enfoques de machine learning aplicados a la predicción del peso corporal animal (Canaza-Cayo et al., 2024).

En este sentido, el uso combinado de múltiples métricas bajo esquemas de validación repetida ha sido recomendado para evitar conclusiones sesgadas y garantizar una evaluación objetiva de los modelos analizados (Canaza-Cayo et al., 2024).

Segunda etapa: Evaluación estadística de las métricas

Con la tabla de resultados ya estructurada, se procedió a realizar un análisis estadístico inferencial que permitiera determinar si existían diferencias significativas en el rendimiento entre los modelos evaluados. Para cada una de las métricas (MAE, MSE, RMSE y R^2), se aplicó un análisis de varianza (ANOVA) de una vía. Esta técnica permitió identificar si las medias de rendimiento de los tres modelos presentaban diferencias estadísticamente significativas, más allá de las fluctuaciones propias del muestreo por pliegues (Firdaus et al., 2024; Paudel et al., 2023).

En los casos donde el ANOVA arrojó resultados significativos, se aplicaron pruebas post hoc, como el test de Tukey HSD, con el fin de precisar entre qué pares de modelos existían dichas diferencias. Este enfoque permitió no solo comparar los modelos a nivel descriptivo, sino también validar sus rendimientos desde una perspectiva estadística rigurosa, lo cual es fundamental para sustentar la elección del modelo más adecuado en aplicaciones de predicción de peso animal mediante machine learning.

Tercera etapa: Interpretación y selección del mejor modelo

Con los resultados del análisis inferencial, se procedió a establecer los criterios que permitirían interpretar las métricas y seleccionar el modelo con mejor desempeño general. Esta decisión no se fundamentó únicamente en los valores promedio obtenidos, sino también en la consistencia de los resultados a lo largo de los pliegues de validación cruzada y en la significancia estadística de las diferencias detectadas.

De acuerdo con lo definido en la etapa metodológica, se priorizó el RMSE como la métrica principal para la comparación entre modelos, por su capacidad para reflejar el impacto de errores grandes en las predicciones, aspecto especialmente crítico en el contexto de estimación de peso en ganado vacuno (Wang et al., 2025; Xu et al., 2024). Aunque también se consideraron métricas complementarias como el MAE y el R^2 , la interpretación y jerarquización de resultados se centró principalmente en el comportamiento del RMSE.

De esta manera, el procedimiento metodológico permitió establecer un marco comparativo riguroso y reproducible para la identificación del modelo más eficiente,

integrando tanto criterios estadísticos como indicadores de precisión técnica, en concordancia con estudios previos en el ámbito agropecuario computacional (Canaza-Cayo et al., 2024; Firdaus et al., 2024).

CAPÍTULO IV RESULTADOS Y DISCUSIÓN

1.1. Resultado específico 1: Construir un conjunto de datos mediante la toma de imágenes de ganado vacuno asociadas a su peso real en un centro de pesaje.

Para alcanzar el primer objetivo específico, se construyó un conjunto de datos supervisado que integró imágenes de perfil de ganado vacuno con su respectivo peso real, obtenido mediante balanza electrónica. Este proceso se ejecutó en cinco etapas secuenciales, desarrolladas en el fundo “Ganadería DR”, distrito de Zapatero, provincia de Lamas.

En la primera etapa, se adecuó el entorno físico. Se instaló una pasarela de conducción y una balanza tipo jaula con sensores Mavin y sistema digital de precisión, permitiendo registrar el peso sin generar estrés en los animales (Figura 11). Este ambiente controlado garantizó la calidad de los registros y la seguridad durante el procedimiento.



Figura 11
Pasarela de conducción del ganado hacia la balanza

La instalación de una pasarela de conducción y una balanza tipo jaula equipada con sensores Mavin y sistema digital de precisión permitió obtener registros de peso con mínima intervención humana y sin generar estrés en los animales. Esta estrategia no solo garantizó la seguridad durante el pesaje, sino también la calidad de los datos obtenidos, en línea con prácticas recomendadas para el desarrollo de sistemas de

monitoreo animal automatizado. Tal como sostienen Arshad et al. (2024), la infraestructura física adecuada es clave para capturar datos confiables que posteriormente alimenten modelos inteligentes en contextos rurales.

Durante la segunda etapa, se pesó a cada ejemplar utilizando la balanza electrónica (Figura 12). A cada animal se le asignó un código numérico único, que luego se usaría para su identificación visual en las imágenes, asegurando la trazabilidad de los datos. Esta información fue registrada en un archivo Excel que contiene el listado completo de los códigos asignados y sus respectivos pesos, el cual se encuentra disponible en el siguiente enlace:

<https://docs.google.com/spreadsheets/d/1cVDNHkCWYMEf3puzwStdg3BgEHcqhP9H/edit?usp=sharing&oid=102905876930999966205&rtpof=true&sd=true>



Figura 12

Ejemplar bovino durante el registro de peso en balanza electrónica

La asignación de un código numérico único a cada ejemplar, enlazado directamente al registro de su peso real, fortaleció la trazabilidad del proceso y permitió una vinculación precisa entre los datos visuales y numéricos. Esta correspondencia fue crucial para la calidad del dataset generado, ya que redujo la posibilidad de errores de emparejamiento, tal como advierten Firdaus et al. (2024) y Zhao et al. (2023), quienes destacan que una mala asociación entre imágenes y datos puede deteriorar significativamente el desempeño predictivo de los modelos.

En la tercera etapa, se colocó un objeto de referencia visual (un círculo de vinil de seis pulgadas) en el cuerpo del animal (Figura 13). Este objeto permitió estandarizar la escala de las imágenes y sirvió como marcador para identificar al ejemplar.



Figura 13
Círculo de vinil colocado para calibración geométrica de imágenes

La colocación de un círculo de vinil de seis pulgadas como objeto de referencia permitió estandarizar la escala visual en todas las imágenes. Esta medida fue especialmente útil para evitar distorsiones en la extracción de características morfológicas, dada la variabilidad en las condiciones de captura. Según X. Li et al. (2023), el uso de objetos geométricos simples como marcadores de escala es una técnica efectiva para entornos sin infraestructura calibrada, y permite asegurar comparabilidad entre muestras.

La cuarta etapa consistió en capturar imágenes del perfil del ganado utilizando un smartphone Samsung Galaxy A52s 5G (Figura 14). Las imágenes se tomaron cuando el animal permanecía inmóvil, procurando una vista lateral clara, en orientación perpendicular a la cámara. Esta técnica flexible respondió a las condiciones de campo y a las necesidades prácticas del entorno rural.



Figura 14
Imagen del ejemplar en vista lateral para análisis morfológico

El uso de un smartphone Samsung Galaxy A52s 5G para la captura de imágenes demostró ser una alternativa viable en contextos rurales, brindando calidad suficiente para la segmentación y análisis posterior. La elección de este dispositivo respondió a criterios de accesibilidad y portabilidad, sin comprometer la nitidez necesaria para la visión computacional. Kumar et al. (2024) ya habían señalado que los teléfonos móviles actuales pueden ser herramientas útiles para la captura de datos visuales en proyectos de bajo costo, siempre que se mantengan ciertas condiciones mínimas de enfoque y contraste.

En la quinta etapa, se organizó y renombró cada imagen según el código visual, vinculándola con su valor de peso registrado (Figura 15). El conjunto final se estructuró con 3500 imágenes, formando una base de datos lista para el entrenamiento de modelos supervisados de regresión. Este conjunto fue almacenado en una carpeta en Google Drive, con el fin de asegurar su preservación y disponibilidad para posteriores fases de análisis y validación.

<https://drive.google.com/drive/folders/1ySTkZRih8lcns9i3gCmzNs2jHTYIV5vg?usp=sharing>

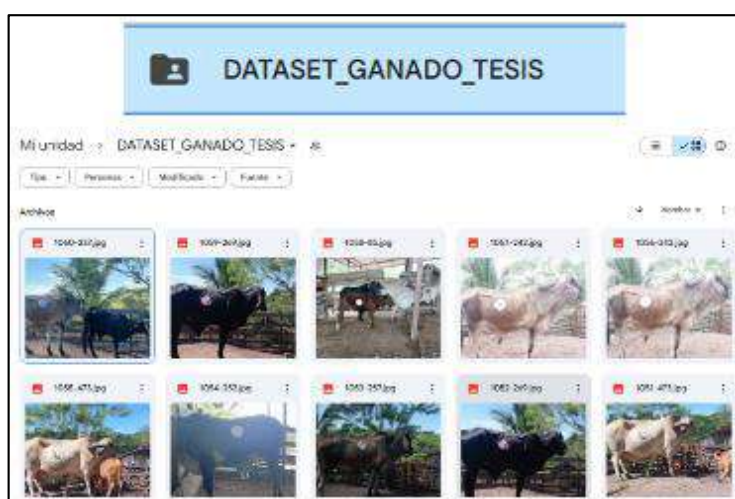


Figura 15
Estructura del dataset organizado en Google Drive

La estructuración de un dataset compuesto por aproximadamente 3500 imágenes permitió conformar una base de datos suficientemente amplia y variada para el entrenamiento de modelos de regresión. Esta cantidad de muestras supera el mínimo sugerido por Bhoj et al. (2022), quienes afirman que una mayor diversidad y volumen de datos mejora la capacidad de generalización de los algoritmos predictivos. Asimismo, la organización sistemática del conjunto en Google Drive aseguró la preservación y disponibilidad de los datos, aspectos esenciales en proyectos que aspiran a ser replicables y escalables.

1.2. Resultado específico 2: Desarrollar modelos híbridos basados en segmentación de imágenes y regresión para estimar de forma no invasiva el peso del ganado vacuno.

Con el propósito de alcanzar el segundo objetivo específico, se desarrolló un flujo metodológico estructurado en seis etapas secuenciales. Este proceso permitió construir modelos híbridos que integran visión computacional y algoritmos de regresión supervisada para estimar el peso corporal del ganado de forma no invasiva. A continuación, se presentan los resultados correspondientes a las cinco primeras etapas del proceso.

En la primera etapa, se cargaron las imágenes del ganado vacuno almacenadas en una carpeta de Google Drive. Cada archivo contenía, en su nombre, un identificador único vinculado al ejemplar y su peso real, obtenido mediante una balanza electrónica. Esta organización permitió asegurar la trazabilidad de los datos desde la fase de preprocesamiento hasta el entrenamiento de los modelos.

```
from google.colab import drive
drive.mount('/content/drive')
```

La organización estructurada de las imágenes y su vinculación explícita con los pesos reales representa un paso esencial en el desarrollo de modelos supervisados. Tal como destacan Arshad et al. (2024) y Löffler et al. (2021), la trazabilidad y estandarización inicial de los datos mejora la calidad del entrenamiento y facilita la automatización de procesos posteriores.

En la segunda etapa, se empleó un modelo YOLOv8 personalizado para detectar automáticamente el círculo de vinil adherido al costado del animal (Figura 16). Este objeto de referencia permitió calcular un factor de escala geométrica a partir del radio detectado en píxeles, con el fin de redimensionar las imágenes y normalizarlas visualmente bajo un mismo estándar. Para ello, se implementó una función programada que procesaba cada imagen, comenzando por la rotación automática en caso de orientación vertical:

```
if img.shape[0] > img.shape[1]:
    print(f"🌀 Rotando imagen vertical: {image_path}")
    img = cv2.rotate(img, cv2.ROTATE_90_COUNTERCLOCKWISE)
```

Luego, se aplicaba el modelo de detección sobre la imagen, extrayendo las coordenadas del círculo:

```
results = model(image_path, imgsiz=1280)
```

```
mask_coords = results[0].masks.xy[0]
```

A partir de los extremos horizontales de la máscara, se calculaba el diámetro del círculo detectado y, con base en un valor objetivo (por ejemplo, 30 píxeles), se obtenía el factor de escala:

```
diameter_detected = x_max - x_min
scale_factor = target_diameter_px / diameter_detected
```

El siguiente paso consistía en redimensionar toda la imagen original utilizando dicho factor, lo que aseguraba que todas las imágenes tuvieran al círculo de referencia con el mismo tamaño, estandarizando así la escala aparente entre animal y cámara.

Este procedimiento automatizado permitió normalizar imágenes de forma consistente, replicando estrategias exitosas utilizadas en otros estudios donde no se dispone de cámaras calibradas ni condiciones controladas.



Figura 16.
Detección y normalización del círculo de referencia

La normalización mediante objetos de referencia físicos ha sido validada en entornos agropecuarios sin cámaras calibradas. Investigaciones como las de X. Li et al. (2023) y Hou et al. (2023) resaltan que el uso de patrones geométricos permite corregir distorsiones debidas a la distancia o al ángulo de captura, asegurando así la comparabilidad entre imágenes.

En la tercera etapa, se aplicó el modelo yolov8x-seg.pt, una variante ligera del algoritmo YOLOv8 orientada a tareas de segmentación semántica. Este modelo permitió identificar automáticamente la clase cow en cada imagen y generar una máscara binaria, en la que se representó la silueta del animal en color rojo sobre un fondo blanco (Figura 17). Esta representación visual estandarizada facilitó la separación del ejemplar

respecto del entorno, lo cual resulta fundamental para las siguientes fases de extracción de características morfológicas.

El procedimiento inició con la lectura de las imágenes previamente normalizadas, sobre las cuales se ejecutó la inferencia del modelo YOLOv8:

```
results = model(input_path, imgsiz=640)[0]
```

Se evaluó la presencia de la clase correspondiente al ganado vacuno identificada por su ID específico en el modelo entrenado, y en caso de detección positiva, se extrajo la máscara de segmentación asociada:

```
if int(cls) == COW_CLASS_ID:  
    mask = results.masks.data[i].cpu().numpy()
```

A partir de esta máscara, se construyó una nueva imagen en la que se pintó la silueta del animal en color rojo, mientras que el fondo se dejó completamente blanco, permitiendo así una separación clara entre figura y contexto:

```
cow_area[:] = COW_COLOR  
cow_mask = cv2.bitwise_and(cow_area, cow_area, mask=mask)  
output_img += cow_mask
```

Finalmente, cada imagen fue redimensionada a una resolución estándar de 256 × 256 píxeles, asegurando la uniformidad del conjunto para el posterior entrenamiento de modelos de regresión:

```
output_img_resized = cv2.resize(output_img, RESOLUCION_OBJETIVO
```

Este proceso fue automatizado para aplicarse de forma masiva sobre todo el conjunto de datos segmentados, generando múltiples copias por ejemplar con fines de balanceo y reforzamiento del dataset. La segmentación automática con YOLOv8 permitió obtener una base visual limpia y coherente, esencial para asegurar la precisión en las siguientes fases del pipeline computacional.

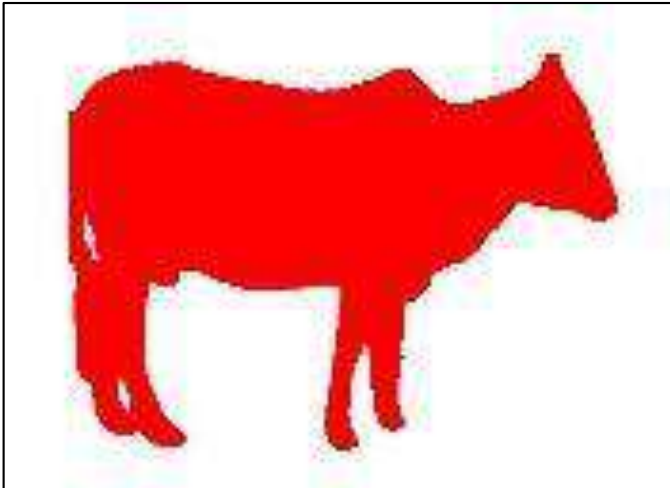


Figura 17
Segmentación del ganado (YOLOv8 preentrenado)

La segmentación precisa del animal constituye un componente esencial en modelos de predicción basados en visión por computadora, ya que permite aislar de manera efectiva las estructuras morfológicas relevantes para el análisis. Estudios como los de Peng et al. (2024) y Devi et al. (2024) han demostrado que modelos preentrenados como YOLOv8, aplicados a tareas de segmentación semántica, pueden identificar la silueta de los ejemplares con alta precisión, sin necesidad de procesos de reentrenamiento local. En el presente estudio, esta capacidad permitió generar máscaras limpias y homogéneas, que sirvieron como base para la posterior extracción de características geométricas, contribuyendo a la robustez del pipeline de predicción.

En la cuarta etapa, se extrajeron automáticamente características morfológicas a partir de las imágenes segmentadas. Estas características incluyeron medidas como área, perímetro, ancho, alto, relación de aspecto y tres Momentos de Hu, todas calculadas a partir del contorno principal detectado en la imagen (Figura 18). Para ello, se desarrolló una función específica que convertía la imagen a escala de grises, aplicaba un umbral binario y localizaba el contorno externo de mayor área:

```
_, binaria = cv2.threshold(gris, 250, 255, cv2.THRESH_BINARY_INV)
contornos, _ = cv2.findContours(binaria, cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)
```

Una vez identificado el contorno principal, se calcularon las métricas morfológicas relevantes:

```
area = cv2.contourArea(contorno_principal)
perimetro = cv2.arcLength(contorno_principal, True)
x, y, ancho, alto = cv2.boundingRect(contorno_principal)
relacion_aspecto = ancho / alto if alto != 0 else 0
```

```
momentos_hu =
cv2.HuMoments(cv2.moments(contorno_principal)).flatten()[:3]
```

Estas variables numéricas constituyeron una representación visual concreta y estandarizada del ejemplar, basada en la forma de su silueta segmentada.



Figura 18
Extracción de características geométricas

Las variables extraídas han sido ampliamente utilizadas en estudios de estimación estructural del cuerpo animal. Autores como Ling et al. (2022) y Xu et al. (2024) destacan la utilidad de estas métricas para capturar la morfología general del ganado y establecer relaciones cuantificables con el peso corporal, especialmente en condiciones no controladas.

En la quinta etapa, cada vector de características geométricas fue vinculado con su peso real correspondiente, información que se extrajo automáticamente desde el nombre del archivo de imagen, codificado previamente durante la organización del dataset:

```
def extraer_peso(nombre_archivo):
    partes = nombre_archivo.split('_')
    try:
        peso = int(partes[4])
        return peso
    except:
        return None
```

Las variables predictoras se almacenaron en la matriz X, mientras que los valores de peso se integraron en el vector y, respetando la correspondencia entre entrada y salida para cada ejemplar. Finalmente, este conjunto de datos supervisado fue serializado y almacenado en un archivo binario mediante la librería joblib, permitiendo su reutilización eficiente en las etapas de entrenamiento y validación:

```
joblib.dump({'X': X, 'y': y}, ruta_datos)
```

Este procedimiento aseguró la coherencia y trazabilidad del conjunto de datos, permitiendo establecer una base sólida para el desarrollo de modelos de regresión confiables.

Tal como afirman Chen et al. (2023) y Canaza-Cayo et al. (2024), la correcta asociación entre vectores de características e información numérica real es fundamental para entrenar modelos de regresión con un alto nivel de precisión. Esta estructura indexada garantiza coherencia en el aprendizaje supervisado y reduce el margen de error en la predicción.

En la sexta etapa, se entrenaron tres modelos híbridos independientes:

- ✓ YOLOv8 + Random Forest
- ✓ YOLOv8 + XGBoost
- ✓ YOLOv8 + LightGBM

Cada modelo fue validado mediante validación cruzada K-Fold con $k = 5$, y sus hiperparámetros fueron ajustados mediante RandomizedSearchCV.

1.2.1. Modelo híbrido YOLOv8 + Random Forest

El primer modelo híbrido desarrollado integró la segmentación automática mediante YOLOv8 y la regresión con el algoritmo Random Forest. El entrenamiento se realizó sobre el conjunto completo de datos geométricos vinculados al peso real de los ejemplares. El modelo alcanzó un MSE de 37.94, un RMSE de 6.16, un MAE de 4.71 y un coeficiente de determinación R^2 de 0.9677. Los hiperparámetros óptimos seleccionados durante la validación fueron: `n_estimators=200`, `min_samples_split=5`, `min_samples_leaf=1`, `max_features='log2'`, `max_depth=None` y `bootstrap=True`.

La evaluación gráfica del modelo Random Forest muestra un comportamiento robusto y confiable en la estimación del peso bovino (Figura 19). La gráfica de dispersión entre predicción y valor real evidencia una fuerte correlación lineal, donde la mayoría de los puntos se alinean cerca de la diagonal ideal, lo que indica alta precisión. La distribución de errores residuales presenta una forma simétrica y centrada en cero, reflejando que los errores son bajos y están equilibrados, sin sesgos significativos. El histograma de errores confirma esta distribución normal, respaldando la consistencia del modelo. Finalmente, la curva de aprendizaje revela una tendencia estable de mejora en la validación conforme aumenta el tamaño del conjunto de entrenamiento, lo que sugiere un modelo bien ajustado, con margen para seguir mejorando su generalización. Estos

indicadores visuales en conjunto validan la eficacia del modelo para tareas de regresión en el contexto agropecuario.

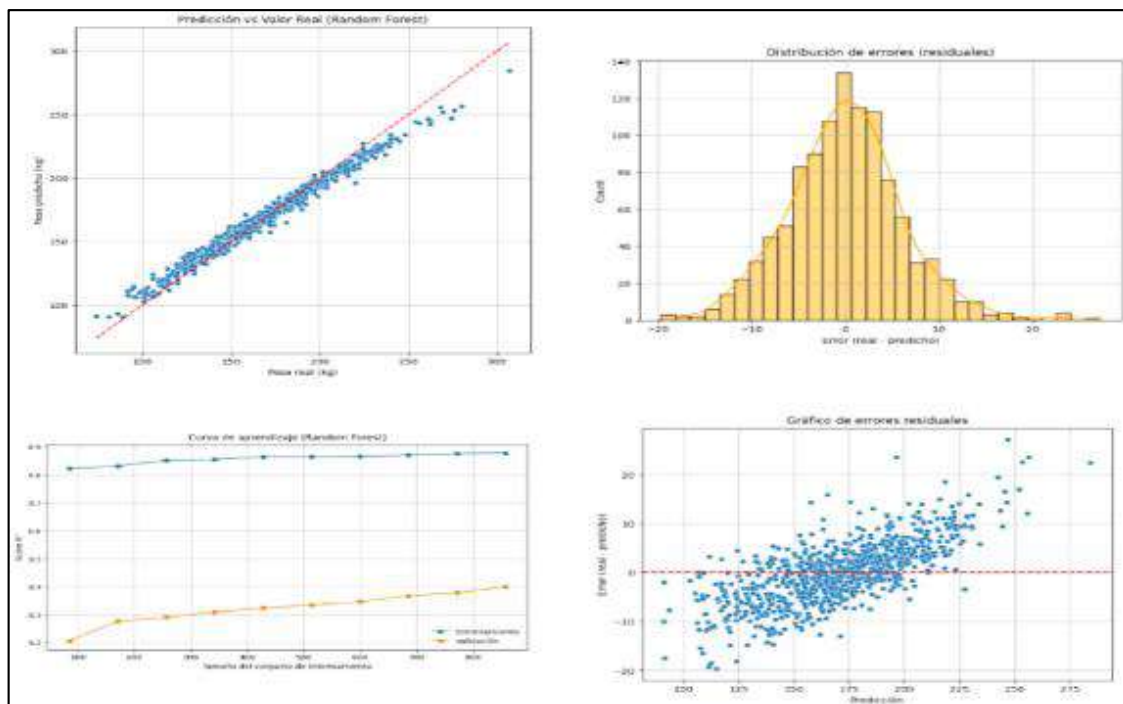


Figura 19

Evaluación gráfica del modelo híbrido YOLOv8 + Random Forest

Desde el enfoque analítico, los resultados posicionan al modelo híbrido YOLOv8 + Random Forest como una opción robusta y confiable para la estimación no invasiva del peso bovino. Firdaus et al. (2024) resaltan su capacidad para manejar datos ruidosos y capturar relaciones no lineales, mientras que Anjar Setiawan et al. (2024) destacan su eficacia en entornos agropecuarios por su baja complejidad y buena interpretabilidad. En este estudio, el modelo alcanzó un R^2 de 0.9677, con errores aceptables (RMSE de 6.16 y MAE de 4.71), validando su aplicabilidad práctica en sistemas de ganadería tecnificada.

1.2.2. Modelo híbrido YOLOv8 + XGBoost

El segundo modelo híbrido integró el proceso de segmentación automática mediante YOLOv8 y el algoritmo de regresión XGBoost. El entrenamiento se realizó sobre el mismo conjunto de datos geométricos, logrando un rendimiento notable con un MSE de 20.88, un RMSE de 4.57, un MAE de 3.37 y un coeficiente de determinación R^2 de 0.9822. Estos indicadores reflejan una alta capacidad predictiva del modelo, sustentada en la adecuada extracción de características morfológicas. Los hiperparámetros óptimos seleccionados durante la validación fueron: $\text{subsample}=0.9$, $\text{reg_lambda}=2$,

reg_alpha=0, n_estimators=300, max_depth=6, learning_rate=0.1, gamma=0 y colsample_bytree=0.9.

La gráfica de predicción versus valor real muestra una alineación estrecha de los puntos respecto a la diagonal, lo que indica alta precisión en las estimaciones. Los errores residuales se distribuyen de manera centrada y simétrica en torno al cero, lo que evidencia una buena calibración sin sesgos relevantes. Esta tendencia también se refleja en el histograma, cuya forma aproximadamente normal respalda la consistencia del modelo. La curva de aprendizaje, por su parte, muestra una mejora progresiva con el aumento del conjunto de entrenamiento, aunque con menor margen de mejora comparativa, lo que sugiere que XGBoost converge rápidamente (**Figura 20**).

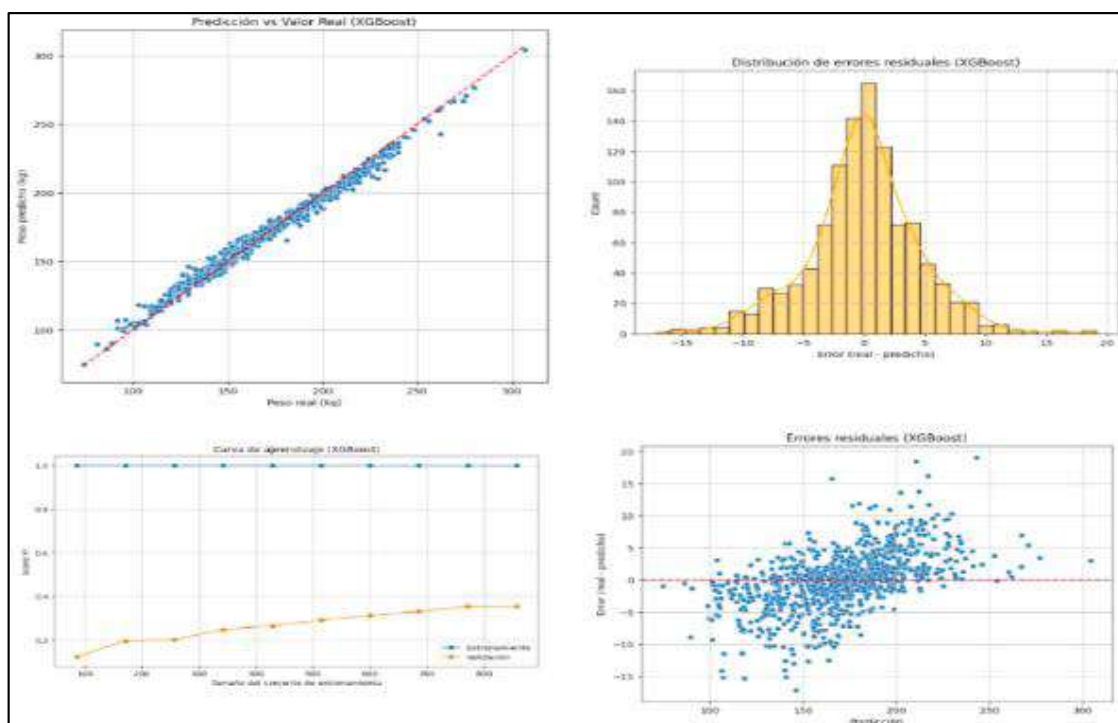


Figura 20

Evaluación gráfica del modelo híbrido YOLOv8 + XGBoost

Desde una perspectiva técnica, los resultados obtenidos validan la idoneidad del modelo XGBoost para tareas de regresión aplicadas a la estimación de peso en ganado vacuno. Tal como lo destacan Afridi et al. (2024) y Z. Liu et al. (2023), su arquitectura basada en árboles secuenciales y su eficaz esquema de regularización permiten capturar relaciones complejas en datos morfológicos. En este estudio, el modelo mostró un desempeño sobresaliente con un RMSE de 4.57 y un R^2 de 0.9822, lo que refleja su capacidad para ajustar las predicciones con alta precisión sin sobreajuste, consolidándose como una alternativa robusta en contextos agropecuarios.

1.2.3. Modelo híbrido YOLOv8 + LightGBM

El tercer modelo híbrido integró la segmentación automática con YOLOv8 y la regresión mediante el algoritmo LightGBM. Este modelo obtuvo un error cuadrático medio (MSE) de 28.39, una raíz del error cuadrático medio (RMSE) de 5.33, un error absoluto medio (MAE) de 4.10 y un coeficiente de determinación R^2 de 0.9758. Aunque estos valores reflejan una capacidad explicativa sólida, las métricas de error fueron más elevadas en comparación con los modelos Random Forest y XGBoost. Los hiperparámetros óptimos seleccionados fueron: `subsample=0.8`, `reg_lambda=2`, `reg_alpha=0.1`, `num_leaves=60`, `n_estimators=400`, `min_child_samples=30`, `max_depth=10`, `learning_rate=0.1` y `colsample_bytree=0.8`.

Las gráficas correspondientes al modelo LightGBM evidencian una correlación positiva entre los valores reales y predichos, con una distribución de puntos cercana a la diagonal ideal (Figura 21). La distribución de errores residuales muestra una forma simétrica y centrada, lo que sugiere ausencia de sesgo sistemático. Sin embargo, el gráfico de dispersión de errores revela una mayor dispersión en los extremos del rango de predicción, indicando una leve disminución de precisión en pesos muy bajos o altos. Finalmente, la curva de aprendizaje refleja una brecha entre entrenamiento y validación, lo que sugiere un posible sobreajuste leve o margen de mejora en la capacidad de generalización.

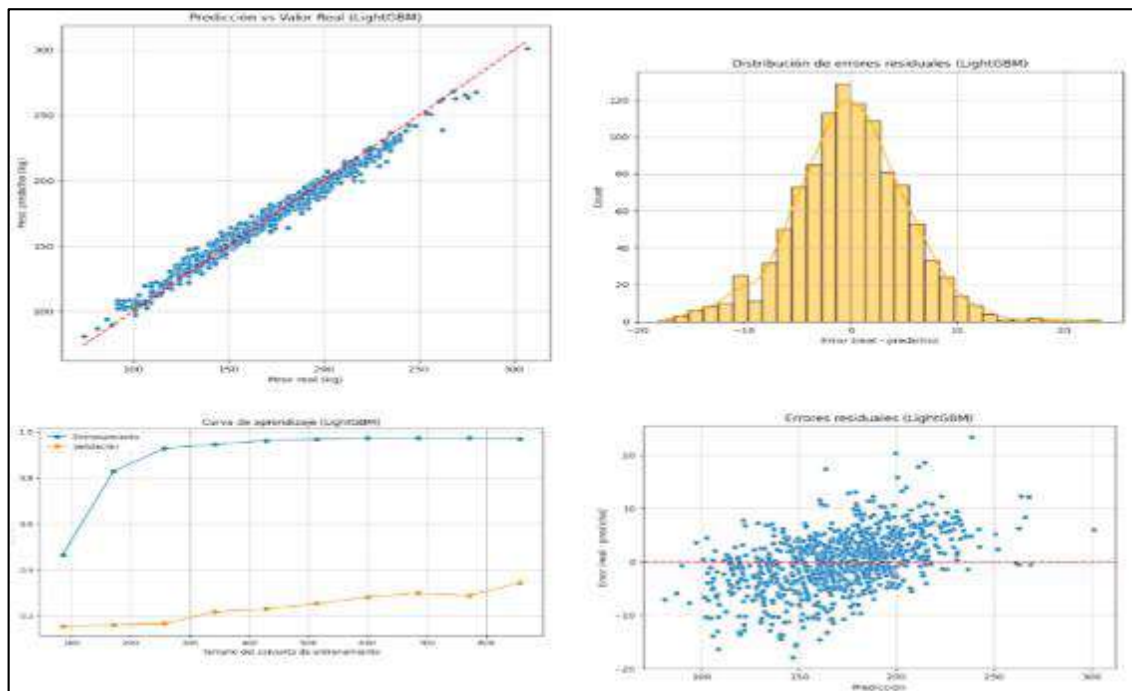


Figura 21

Evaluación gráfica del modelo híbrido YOLOv8 + LightGBM Regressor

Desde el enfoque técnico, LightGBM demostró ser un modelo competente en la estimación del peso bovino, aunque su rendimiento fue inferior al de sus homólogos. Tal como advierten Hou et al. (2023) y Xu et al. (2024), este algoritmo, si bien es altamente eficiente en términos computacionales, puede ser sensible a configuraciones subóptimas de hiperparámetros como `num_leaves` o los factores de regularización. En este estudio, a pesar de alcanzar un R^2 elevado (0.9758), se observaron errores relativamente más altos (MAE de 4.10) y una mayor dispersión en la predicción frente a morfologías variables, lo cual sugiere una menor capacidad de generalización ante la heterogeneidad estructural del ganado.

1.3. Resultado específico 3: métricas de rendimiento de los modelos híbridos

Con el propósito de alcanzar el tercer objetivo específico de esta investigación, se diseñó un procedimiento riguroso que permitió evaluar y comparar el rendimiento de tres modelos híbridos construidos previamente. Estos modelos combinan el algoritmo YOLOv8 con tres técnicas de regresión distintas: Random Forest (RF), XGBoost (XGB) y LightGBM (LGBM). El proceso de evaluación se estructuró en tres fases consecutivas: primero, la consolidación de resultados en tabla comparativa; segundo, evaluación estadística de las métricas; y tercero, la interpretación y selección del mejor modelo.

Para iniciar el proceso, se utilizó un conjunto de datos de prueba previamente reservado, el cual fue segmentado en dieciséis subconjuntos. Esta segmentación permitió probar cada modelo en diferentes particiones del conjunto de datos, asegurando así una evaluación más confiable de su rendimiento.

```
kf = KFold(n_splits=K, shuffle=False)
indices = np.arange(len(X))
folds = list(kf.split(indices))
```

En cada una de las dieciséis iteraciones se excluyó un subconjunto distinto del conjunto de prueba, no con la finalidad de utilizarlo como conjunto de validación, sino para introducir variabilidad y evitar sesgos en el proceso de evaluación. Al eliminar sistemáticamente una parte diferente del dataset en cada corrida, se logró generar múltiples escenarios de prueba independientes, sobre los cuales se aplicaron directamente los modelos previamente entrenados. Las predicciones se realizaron únicamente sobre los datos restantes de cada corrida, manteniendo constante el modelo y modificando únicamente la muestra sobre la que se evaluaba su desempeño.

```
mask = np.ones(len(X), dtype=bool)
mask[val_idx] = False # Se excluye una parte distinta en cada iteración
```

```
X_corrida = X[mask]
y_corrida = y[mask]
```

Una vez obtenidas las métricas en las dieciséis corridas, los resultados fueron organizados en una hoja de cálculo, permitiendo visualizar y comparar el desempeño de los modelos de forma estructurada (**Tabla 3**). Esto permitió observar tendencias generales y diferencias de rendimiento entre los tres enfoques híbridos evaluados.

```
promedios = df_resultados.drop(columns='Corrida').mean().to_dict()
df_resultados = pd.concat([df_resultados,
pd.DataFrame([promedios]), ignore_index=True)
```

Tabla 3
Métricas según 16 iteraciones

MODELOS	GRUPO	R ²	MAE	RMSE
YOLOv8 + Random Forest	1	0.9676	4.7749	6.2369
	2	0.9679	4.673	6.1191
	3	0.9678	4.6892	6.1276
	4	0.9674	4.7601	6.2281
	5	0.9681	4.6761	6.1001
	6	0.9669	4.7402	6.2049
	7	0.9681	4.6920	6.1396
	8	0.9676	4.7131	6.1871
	9	0.9677	4.7089	6.1770
	10	0.9676	4.7146	6.1448
	11	0.9679	4.6813	6.1146
	12	0.9683	4.6802	6.0988
	13	0.9677	4.7111	6.1648
	14	0.9674	4.7160	6.1662
	15	0.9670	4.7086	6.1621
	16	0.9679	4.7139	6.1762
YOLOv8 + XGBoost	1	0.9823	3.3982	4.6103
	2	0.9824	3.344	4.5347
	3	0.9821	3.3791	4.5686
	4	0.9825	3.3739	4.5603
	5	0.9824	3.3473	4.5383
	6	0.9818	3.3882	4.6000
	7	0.9825	3.3500	4.5511
	8	0.9823	3.3835	4.5737
	9	0.9826	3.3304	4.5288
	10	0.9816	3.4097	4.6294
	11	0.9823	3.3393	4.5415
	12	0.9824	3.3541	4.5519
	13	0.9824	3.3611	4.5543
	14	0.9818	3.4012	4.6093
	15	0.9818	3.3615	4.5736
	16	0.9823	3.3889	4.5833
YOLOv8 + LightGBM	1	0.9758	4.1282	5.3841
	2	0.9767	4.0231	5.2188
	3	0.9757	4.1096	5.3663
	4	0.9758	4.1419	5.3267
	5	0.9759	4.0863	5.3041
	6	0.9752	4.1159	5.3714
	7	0.9762	4.0699	5.3014
	8	0.9758	4.1295	5.3550
	9	0.9761	4.0868	5.3086
	10	0.9752	4.1453	5.3761
	11	0.9762	4.0668	5.2739
	12	0.9763	4.0735	5.2759
	13	0.9764	4.0718	5.2725
	14	0.9754	4.1141	5.3517
	15	0.9750	4.1249	5.3694
	16	0.9756	4.1563	5.3850

La evaluación cruzada con dieciséis subconjuntos permitió observar el comportamiento de cada modelo sin necesidad de reentrenamiento, introduciendo variabilidad controlada para una medición más objetiva. Los tres modelos híbridos mostraron distintos patrones de estabilidad y precisión: YOLOv8 + Random Forest evidenció resultados consistentes entre grupos; YOLOv8 + XGBoost mostró métricas de error relativamente compactas y valores altos de R^2 ; mientras que YOLOv8 + LightGBM presentó mayor dispersión en las métricas, lo que sugiere una sensibilidad particular a la composición de los subconjuntos.

Esta estrategia de evaluación se alinea con lo recomendado por Firdaus et al. (2024), quienes proponen el uso de pruebas múltiples para validar modelos con mayor solidez metodológica. Además, Arshad et al. (2024) señalan que los algoritmos basados en árboles, como Random Forest y XGBoost, tienden a mantener un rendimiento estable en entornos con alta variabilidad estructural, como ocurre con datos morfológicos de ganado. Por otro lado, las observaciones de Z. Liu et al. (2023) respecto a la sensibilidad de LightGBM ante la configuración hiperparamétrica encuentran eco en los resultados aquí obtenidos, donde se evidencia cierta inestabilidad en sus métricas. En conjunto, este análisis permite caracterizar con mayor detalle el comportamiento predictivo de cada arquitectura híbrida bajo condiciones reales y controladas.

Para comparar el rendimiento de los modelos híbridos evaluados, se eligió el RMSE como métrica principal, dado que penaliza con mayor severidad los errores grandes, aspecto crítico en la estimación del peso del ganado. Esta elección permite una comparación más rigurosa entre modelos, especialmente en escenarios donde una predicción inexacta puede afectar decisiones productivas. Además, el RMSE es ampliamente reconocido en la literatura especializada como referencia estándar para la evaluación de modelos de regresión en entornos agrícolas (Wang et al., 2025; Xu et al., 2024). Con el propósito de determinar si existen diferencias significativas en el desempeño de los modelos híbridos, se plantean las siguientes hipótesis:

- **H₀**: No existen diferencias significativas entre los modelos empleados.
- **H_α**: Al menos uno de los modelos presenta un desempeño significativamente superior.

Prueba de homogeneidad de varianza

A continuación, se presenta los resultados de la prueba, con el fin de evaluar si las varianzas de los modelos empleados son similares o diferentes (**Tabla 4**).

Tabla 4
Prueba de homogeneidad de varianza

		Estadístico de Levene	gl1	gl2	Sig.
RMSE	Se basa en la media	2.750	2	45	.075
	Se basa en la mediana	2.404	2	45	.102
	Se basa en la mediana y con gl ajustado	2.404	2	40.735	.103
	Se basa en la media recortada	2.766	2	45	.074

La Tabla 4 de pruebas de homogeneidad de varianzas mediante el test de Levene indica que no se encontraron diferencias estadísticamente significativas en las varianzas del RMSE entre los tres modelos híbridos evaluados, ya que todos los valores de significancia (Sig.) superan el umbral de 0.05. Esto sugiere que se cumple el supuesto de homocedasticidad, requisito fundamental para la aplicación válida del análisis de varianza (ANOVA) en la comparación de medias.

Posteriormente, en la segunda etapa, se aplicó un análisis de varianza (ANOVA) de una vía con el fin de verificar si existían diferencias significativas en el rendimiento de los modelos híbridos evaluados (**Tabla 5**).

Tabla 5
Resultados de la prueba ANOVA

	RMSE				
	Suma de cuadrados	gl	Media cuadrática	F	Sig.
Entre grupos	20.237	2	10.119	5866.257	<.001
Dentro de grupos	.078	45	.002		
Total	20.315	47			

Los resultados del análisis de varianza (ANOVA) muestran que existen diferencias estadísticamente significativas en los valores de RMSE entre los modelos híbridos evaluados, ya que el valor de significancia es menor a 0.001 ($p < 0.05$). Esto indica que se rechaza la hipótesis nula (H_0), lo que implica que al menos uno de los modelos presenta un desempeño significativamente distinto en términos de error cuadrático medio. El alto valor del estadístico F (5866.257) respalda esta conclusión, evidenciando una diferencia sustancial en la media de los errores entre los grupos.

Dado que el análisis de varianza (ANOVA) evidenció diferencias estadísticamente significativas en el desempeño de los modelos híbridos evaluados, se hace necesario aplicar una prueba post hoc para identificar específicamente entre qué pares de modelos se presentan dichas diferencias. Para ello, se empleará la prueba de comparaciones múltiples de Tukey HSD, la cual resulta adecuada considerando que se cumplió con el supuesto de homogeneidad de varianzas (**Tabla 6**). Esta prueba permitirá precisar con mayor detalle cuáles combinaciones de algoritmos presentan ventajas significativas en términos de su error cuadrático medio (RMSE), consolidando así la validez comparativa de los resultados obtenidos.

Tabla 6
Prueba de Tukey para comparación del RMSE entre modelos híbridos

Variable dependiente: RMSE						
HSD Tukey						
(I) Modelo	(J) Modelo	Diferencia de medias (I-J)	Error estándar	Sig.	Intervalo de confianza al 95%	
					Límite inferior	Límite superior
YOLOv8 + Random Forest	YOLOv8 + XGBoost	1.5899250*	.0146837	<.001	1.554337	1.625513
	YOLOv8 + LightGBM	.8316875*	.0146837	<.001	.796100	.867275
YOLOv8 + XGBoost	YOLOv8 + Random Forest	-1.5899250*	.0146837	<.001	-1.625513	-1.554337
	YOLOv8 + LightGBM	-.7582375*	.0146837	<.001	-.793825	-.722650
YOLOv8 + LightGBM	YOLOv8 + Random Forest	-.8316875*	.0146837	<.001	-.867275	-.796100
	YOLOv8 + XGBoost	.7582375*	.0146837	<.001	.722650	.793825

La prueba de Tukey HSD reveló diferencias estadísticamente significativas en el RMSE entre todos los pares de modelos híbridos evaluados. En cada comparación, el valor de significancia fue menor a 0.001, lo que indica que las discrepancias observadas en el rendimiento no son atribuibles al azar. Por ejemplo, la diferencia media entre YOLOv8 + Random Forest y YOLOv8 + XGBoost fue de 1.5899, con un intervalo de confianza que no incluye el cero, consolidando la existencia de una diferencia real. De igual manera, las combinaciones YOLOv8 + LightGBM también mostraron diferencias significativas tanto frente a Random Forest como a XGBoost. En conjunto, estos resultados permiten afirmar que cada modelo presenta un comportamiento predictivo distinto en cuanto al error cuadrático medio, justificando la necesidad de seleccionar el modelo más adecuado según criterios de precisión y robustez.

Posteriormente en la tercera etapa, con el propósito de complementar el análisis inferencial, se presentan los estadísticos descriptivos del RMSE para cada uno de los modelos híbridos evaluados, lo cual permite observar el comportamiento promedio y la dispersión del error en cada grupo antes de aplicar la prueba de comparaciones múltiples (Tabla 7).

Tabla 7
Estadísticos descriptivos

RMSE	N	Media	Desv. estándar	Error estándar	95% de intervalo de confianza para la media			
					Límite inferior	Límite superior	Mínimo	Máximo
YOLOv8 + Random Forest	16	6.159244	.0421697	.0105424	6.136773	6.181714	6.0988	6.2369
YOLOv8 + XGBoost	16	4.569319	.0300900	.0075225	4.553285	4.585353	4.5288	4.6294
YOLOv8 + LightGBM	16	5.327556	.0499097	.0124774	5.300961	5.354151	5.2188	5.3850
Total	48	5.352040	.6574434	.0948938	5.161138	5.542941	4.5288	6.2369

Los resultados muestran que el modelo YOLOv8 + Random Forest registró el mayor valor promedio de RMSE (6.159), indicando un mayor error en la predicción del peso

bovino respecto a los otros modelos. En contraste, el modelo YOLOv8 + XGBoost obtuvo el menor RMSE promedio (4.569), seguido por YOLOv8 + LightGBM (5.327). Las desviaciones estándar son bajas en los tres casos, lo que evidencia una distribución consistente de los errores dentro de cada grupo. Esta tendencia sugiere diferencias marcadas en el desempeño predictivo entre los modelos, lo que será confirmado en el análisis de comparación de medias posterior.

Con el fin de facilitar la visualización comparativa del desempeño de los modelos híbridos analizados, se presenta el gráfico de medias del RMSE para cada uno de los algoritmos evaluados (Figura 1). Esta representación gráfica permite identificar, de forma intuitiva, las diferencias en los errores de predicción promedio asociados a cada modelo.

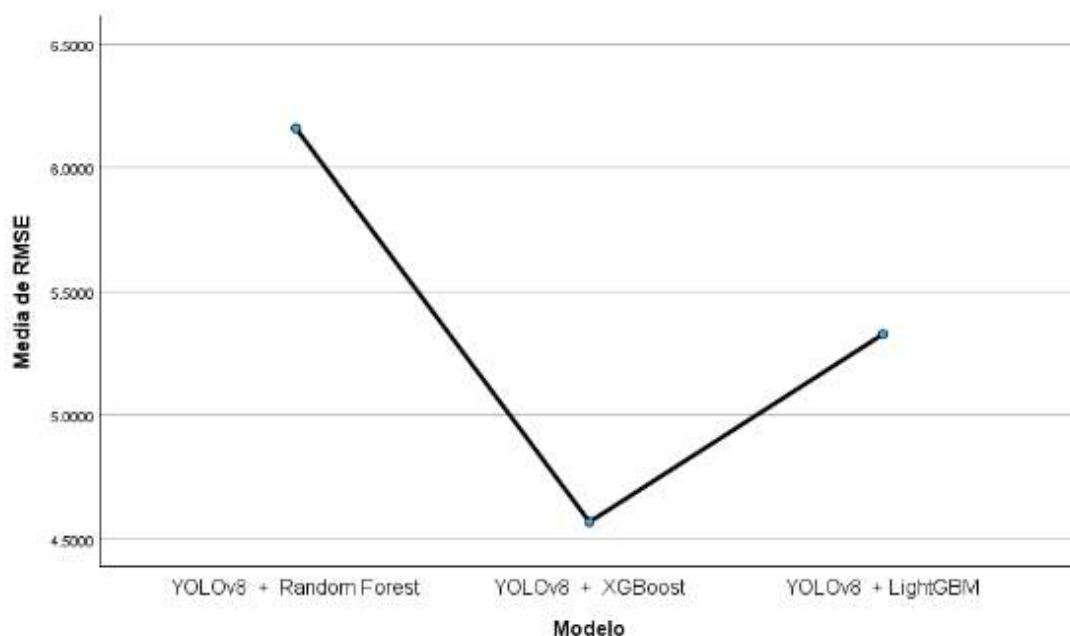


Figura 22
Gráficos de medias

El gráfico evidencia que el modelo YOLOv8 combinado con Random Forest presenta la mayor media de RMSE, mientras que la combinación con XGBoost obtiene el valor más bajo, seguido por el modelo con LightGBM. Esta variabilidad sugiere una diferencia sustancial en la precisión predictiva de los modelos híbridos frente a la estimación del peso del ganado vacuno.

Los resultados reflejados en la Figura 22 revelan diferencias notables en el desempeño de los modelos híbridos evaluados, particularmente en relación al RMSE promedio alcanzado por cada uno. El modelo YOLOv8 + XGBoost obtuvo el menor valor de RMSE, seguido por YOLOv8 + LightGBM, mientras que YOLOv8 + Random Forest

presentó el valor más alto. Esta jerarquía en el rendimiento es consistente con lo reportado por Peng et al. (2024), quienes validaron la efectividad de YOLOv8 en tareas de predicción en condiciones adversas, resaltando su precisión al combinarse con técnicas avanzadas de regresión.

Por otro lado, Canaza-Cayo et al. (2024) destacaron el buen desempeño de Random Forest en escenarios locales, sin embargo, los resultados obtenidos en esta investigación indican que, si bien Random Forest puede ser robusto, su desempeño es más sensible ante la variabilidad morfológica del ganado vacuno, lo cual podría explicar su RMSE más elevado. De forma similar, Hou et al. (2023) y Xiong et al. (2023) subrayaron la importancia de capturar características tridimensionales detalladas, algo que XGBoost habría aprovechado mejor al procesar las salidas de YOLOv8, logrando mayor precisión.

En ese sentido, los resultados reafirman que, aunque todos los modelos híbridos evaluados son técnicamente viables, las diferencias en su desempeño son significativas, y responden tanto a la naturaleza del algoritmo de regresión como a su capacidad para adaptarse a la heterogeneidad del ganado. Esta evidencia respalda la idea de que la combinación de segmentación profunda con regresores optimizados puede potenciar la precisión en sistemas de pesaje no invasivo, alineándose con las tendencias actuales en inteligencia artificial aplicada a la ganadería (Ormeño-Ayala & Zapata-Ttito, 2024; Ruchay et al., 2022).

CONCLUSIONES

La construcción del conjunto de datos mediante la toma de imágenes laterales de ganado vacuno, asociadas a su peso real, permitió generar una base robusta y trazable. La implementación de un protocolo de captura estandarizado, con un objeto de referencia visual y una balanza electrónica calibrada, aseguró la calidad, coherencia y fiabilidad de los datos obtenidos. Esta infraestructura de datos constituye un insumo clave para el entrenamiento y evaluación de modelos predictivos no invasivos en contextos rurales.

Los modelos híbridos desarrollados, que integraron la segmentación automática de imágenes mediante YOLOv8 con algoritmos de regresión como XGBoost, Random Forest y LightGBM, demostraron ser técnicamente viables para estimar de forma precisa el peso del ganado. La estrategia de extracción de características morfológicas relevantes (área, perímetro, bounding box y momentos de Hu) permitió reducir la complejidad visual sin perder capacidad predictiva.

Las métricas de rendimiento aplicadas (MAE, RMSE y R^2), evaluadas bajo un esquema de validación cruzada con pruebas estadísticas como ANOVA y Tukey HSD, revelaron diferencias significativas entre los modelos híbridos comparados. El modelo YOLOv8 + XGBoost obtuvo el menor error promedio (RMSE) y el mayor coeficiente de determinación, consolidándose como el algoritmo más eficiente para el pesaje no invasivo del ganado vacuno.

Se concluye que la combinación de técnicas de visión computacional y algoritmos de regresión constituye una solución efectiva, ética y accesible para el pesaje no invasivo del ganado. La propuesta metodológica, validada en un entorno real de producción ganadera en San Martín, ofrece una alternativa moderna y replicable que contribuye a la mejora del bienestar animal, la eficiencia operativa y la innovación tecnológica en el sector pecuario.

RECOMENDACIONES

Se recomienda ampliar el conjunto de datos con imágenes capturadas en distintas condiciones de luz, raza y textura corporal del ganado, a fin de fortalecer la capacidad generalizable de los modelos entrenados y asegurar su aplicabilidad en otros escenarios rurales del país.

Para futuras implementaciones, se sugiere explorar modelos de regresión más avanzados, como redes neuronales profundas (DNN) o enfoques basados en aprendizaje por transferencia (transfer learning), que podrían mejorar aún más la precisión en la predicción del peso y adaptarse a otros fines zootécnicos.

Se recomienda incorporar nuevas métricas de evaluación como el MAPE (Error Porcentual Absoluto Medio) y realizar pruebas de campo longitudinales que midan la estabilidad del modelo ante variaciones temporales, permitiendo validar su desempeño sostenido a lo largo del tiempo.

Es recomendable promover la adopción de esta tecnología en asociaciones de pequeños y medianos ganaderos de la región San Martín, mediante talleres de capacitación y asesoría técnica, así como articular con instituciones públicas para el financiamiento de soluciones tecnológicas inclusivas que fortalezcan la transformación digital del agro.

REFERENCIAS BIBLIOGRÁFICAS

- Afridi, H., Ullah, M., Nordbø, Ø., Hoff, S. C., Furre, S., Larsgard, A. G., & Cheikh, F. A. (2024). Analyzing Data Modalities for Cattle Weight Estimation Using Deep Learning Models. *Journal of Imaging*, 10(3), 72. <https://doi.org/10.3390/jimaging10030072>
- Anjar Setiawan, Ema Utami, & Dhani Ariatmanto. (2024). Cattle Weight Estimation Using Linear Regression and Random Forest Regressor. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, 8(1), 72–79. <https://doi.org/10.29207/resti.v8i1.5494>
- Arshad, M. F., Burrai, G. Pietro, Varcasia, A., Sini, M. F., Ahmed, F., Lai, G., Polinas, M., Antuofermo, E., Tamponi, C., Cocco, R., Corda, A., & Parpaglia, M. L. P. (2024). The groundbreaking impact of digitalization and artificial intelligence in sheep farming. *Research in Veterinary Science*, 170, 105197. <https://doi.org/10.1016/j.rvsc.2024.105197>
- Bhoj, S., Tarafdar, A., Chauhan, A., Singh, M., & Gaur, G. K. (2022). Image processing strategies for pig liveweight measurement: Updates and challenges. *Computers and Electronics in Agriculture*, 193, 106693. <https://doi.org/10.1016/j.compag.2022.106693>
- Bi, Y., Campos, L. M., Wang, J., Yu, H., Hanigan, M. D., & Morota, G. (2023). Depth video data-enabled predictions of longitudinal dairy cow body weight using thresholding and Mask R-CNN algorithms. *Smart Agricultural Technology*, 6, 100352. <https://doi.org/10.1016/j.atech.2023.100352>
- Brando, S., & Norman, M. (2023). Handling and Training of Wild Animals: Evidence and Ethics-Based Approaches and Best Practices in the Modern Zoo. *Animals*, 13(14), 2247. <https://doi.org/10.3390/ani13142247>
- Canaza-Cayo, A. W., Churata-Huacani, R., Çakmakçı, C., Rodríguez-Huanca, F. H., de Sousa Bueno Filho, J. S., Fernandes, T. J., & Cruz, Y. C. R. D. La. (2024). Use of machine learning approaches for body weight prediction in Peruvian Corriedale Sheep. *Smart Agricultural Technology*, 7, 100419. <https://doi.org/10.1016/j.atech.2024.100419>
- Chen, H., Liang, Y., Huang, H., Huang, Q., Gu, W., & Liang, H. (2023). Live Pig-Weight Learning and Prediction Method Based on a Multilayer RBF Network. *Agriculture*, 13(2), 253. <https://doi.org/10.3390/agriculture13020253>

- da Cunha, A. C. R., Antunes, R. C., da Costa, W. G., Rebouças, G. F., Leite, C. D. S., & do Carmo, A. S. (2024). Body weight prediction in crossbred pigs from digital images using computer vision. *Livestock Science*, 282, 105433. <https://doi.org/10.1016/j.livsci.2024.105433>
- Deng, H., Yang, G., Xu, X., Hua, Z., Liu, J., & Song, H. (2025). Fusion of CREStereo and MobileViT-Pose for rapid measurement of cattle body size. *Computers and Electronics in Agriculture*, 232, 110103. <https://doi.org/10.1016/j.compag.2025.110103>
- Devi, I., Singh, N., Dudi, K., Ranjan, R., Lathwal, S. S., Tomar, D. S., & Nagar, H. (2024). Deep learning aided computer vision system for automated linear type trait evaluation in dairy cows. *Smart Agricultural Technology*, 8, 100509. <https://doi.org/10.1016/j.atech.2024.100509>
- Firdaus, F., Atmoko, B., Ibrahim, A., Nugroho, T., Baliarti, E., & Panjono, P. (2024). A first meta-analysis study on body weight prediction method for beef cattle based on digital image processing. *Journal of Advanced Veterinary and Animal Research*, (0), 1. <https://doi.org/10.5455/javar.2024.k760>
- Hou, Z., Huang, L., Zhang, Q., & Miao, Y. (2023). Body weight estimation of beef cattle with 3D deep learning model: PointNet++. *Computers and Electronics in Agriculture*, 213, 108184. <https://doi.org/10.1016/j.compag.2023.108184>
- Hou, Z., Zhang, Q., Zhang, B., Zhang, H., Huang, L., & Wang, M. (2025). CattlePartNet: An identification approach for key region of body size and its application on body measurement of beef cattle. *Computers and Electronics in Agriculture*, 232, 110013. <https://doi.org/10.1016/j.compag.2025.110013>
- Jiang, S., Zhang, G., Shen, Z., Zhong, P., Tan, J., & Liu, J. (2024). Pig Weight Estimation Method Based on a Framework Combining Mask R-CNN and Ensemble Regression Model. *Animals*, 14(14), 2122. <https://doi.org/10.3390/ani14142122>
- KOJIMA, T., OISHI, K., AOKI, N., MATSUBARA, Y., UETE, T., FUKUSHIMA, Y., INOUE, G., SATO, S., SHIRAIISHI, T., HIROOKA, H., & MASUDA, T. (2022). Estimation of beef cow body condition score: a machine learning approach using three-dimensional image data and a simple approach with heart girth measurements. *Livestock Science*, 256, 104816. <https://doi.org/10.1016/j.livsci.2021.104816>

- Kumar, T., Brennan, R., Mileo, A., & Bendeche, M. (2024). Image Data Augmentation Approaches: A Comprehensive Survey and Future Directions. *IEEE Access*, 12, 187536–187571. <https://doi.org/10.1109/ACCESS.2024.3470122>
- Li, G., Huang, Y., Chen, Z., Chesser, G. D., Purswell, J. L., Linhoss, J., & Zhao, Y. (2021). Practices and Applications of Convolutional Neural Network-Based Computer Vision Systems in Animal Farming: A Review. *Sensors*, 21(4), 1492. <https://doi.org/10.3390/s21041492>
- Li, G., Liu, X., Ma, Y., Wang, B., Zheng, L., & Wang, M. (2022). Body size measurement and live body weight estimation for pigs based on back surface point clouds. *Biosystems Engineering*, 218, 10–22. <https://doi.org/10.1016/j.biosystemseng.2022.03.014>
- Li, J., Ma, W., Bai, Q., Tulpan, D., Gong, M., Sun, Y., Xue, X., Zhao, C., & Li, Q. (2023). A posture-based measurement adjustment method for improving the accuracy of beef cattle body size measurement based on point cloud data. *Biosystems Engineering*, 230, 171–190. <https://doi.org/10.1016/j.biosystemseng.2023.04.014>
- Li, R., Wen, Y., Zhang, S., Xu, X., Ma, B., & Song, H. (2024). Automated measurement of beef cattle body size via key point detection and monocular depth estimation. *Expert Systems with Applications*, 244, 123042. <https://doi.org/10.1016/j.eswa.2023.123042>
- Li, X., Wu, J., Zhao, Z., Zhuang, Y., Sun, S., Xie, H., Gao, Y., & Xiao, D. (2023). An Improved Method for Broiler Weight Estimation Integrating Multi-Feature with Gradient Boosting Decision Tree. *Animals*, 13(23), 3721. <https://doi.org/10.3390/ani13233721>
- Ling, Y., Jimin, Z., Caixing, L., Xuhong, T., & Sumin, Z. (2022). Point cloud-based pig body size measurement featured by standard and non-standard postures. *Computers and Electronics in Agriculture*, 199, 107135. <https://doi.org/10.1016/j.compag.2022.107135>
- Liu, J., Wang, J., Xiong, J., Cheng, W., Sun, H., Yong, Z., & Wang, N. (2021). Hybrid Models Incorporating Bivariate Statistics and Machine Learning Methods for Flash Flood Susceptibility Assessment Based on Remote Sensing Datasets. *Remote Sensing*, 13(23), 4945. <https://doi.org/10.3390/rs13234945>

- Liu, J., Xiao, D., Liu, Y., & Huang, Y. (2023). A Pig Mass Estimation Model Based on Deep Learning without Constraint. *Animals*, 13(8), 1376. <https://doi.org/10.3390/ani13081376>
- Liu, Z., Hua, J., Xue, H., Tian, H., Chen, Y., & Liu, H. (2023). Body Weight Estimation for Pigs Based on 3D Hybrid Filter and Convolutional Neural Network. *Sensors*, 23(18), 7730. <https://doi.org/10.3390/s23187730>
- Löffler, F., Wesp, V., König-Ries, B., & Klan, F. (2021). Dataset search in biodiversity research: Do metadata in data repositories reflect scholarly information needs? *PLOS ONE*, 16(3), e0246099. <https://doi.org/10.1371/journal.pone.0246099>
- Los, S., Mücher, C. A., Kramer, H., Franke, G. J., & Kamphuis, C. (2023). Estimating body dimensions and weight of cattle on pasture with 3D models from UAV imagery. *Smart Agricultural Technology*, 4, 100167. <https://doi.org/10.1016/j.atech.2022.100167>
- Lozano, R. R. (2020). *Evaluación de la condición corporal y el rendimiento de la canal de los bovinos faenados en el camal privado Bello Horizonte—distrito de La Banda de Shilcayo—San Martín*. <http://hdl.handle.net/11458/4592>
- Ma, W., Qi, X., Sun, Y., Gao, R., Ding, L., Wang, R., Peng, C., Zhang, J., Wu, J., Xu, Z., Li, M., Zhao, H., Huang, S., & Li, Q. (2024). Computer Vision-Based Measurement Techniques for Livestock Body Dimension and Weight: A Review. *Agriculture*, 14(2), 306. <https://doi.org/10.3390/agriculture14020306>
- Nguyen, A. H., Holt, J. P., Knauer, M. T., Abner, V. A., Lobaton, E. J., & Young, S. N. (2023). Towards rapid weight assessment of finishing pigs using a handheld, mobile RGB-D camera. *Biosystems Engineering*, 226, 155–168. <https://doi.org/10.1016/j.biosystemseng.2023.01.005>
- Ogawa, N. N., Silva, G. L., da Costa Barbon, A. P. A., da Costa Flaiban, K. K. M., da Silva, C. A., Rocha, L. M., & Bridi, A. M. (2024). Animal Welfare Assessment and Meat Quality through Assessment of Stress Biomarkers in Fattening Pigs with and without Visible Damage during Slaughter. *Animals*, 14(5), 700. <https://doi.org/10.3390/ani14050700>
- Ormeño-Ayala, Y. I., & Zapata-Ttito, A. G. (2024). Aplicación de la red convolucional Mask R-CNN para la estimación del peso corporal del cuy. *Revista Científica de Sistemas e Informática*, 4(1), e614. <https://doi.org/10.51252/rcsi.v4i1.614>

- Paudel, S., de Sousa, R. V., Sharma, S. R., & Brown-Brandl, T. (2023). Deep Learning Models to Predict Finishing Pig Weight Using Point Clouds. *Animals*, *14*(1), 31. <https://doi.org/10.3390/ani14010031>
- Peng, Y., Peng, Z., Zou, H., Liu, M., Hu, R., Xiao, J., Liao, H., Yang, Y., Huo, L., & Wang, Z. (2024). A dynamic individual method for yak heifer live body weight estimation using the YOLOv8 network and body parameter detection algorithm. *Journal of Dairy Science*, *107*(8), 6178–6191. <https://doi.org/10.3168/jds.2023-24065>
- Ruchay, A., Kober, V., Dorofeev, K., Kolpakov, V., Gladkov, A., & Guo, H. (2022). Live Weight Prediction of Cattle Based on Deep Regression of RGB-D Images. *Agriculture*, *12*(11), 1794. <https://doi.org/10.3390/agriculture12111794>
- Temenos, A., Voulodimos, A., Korelidou, V., Gelasakis, A., Kalogeras, D., Doulamis, A., & Doulamis, N. (2024). Goat-CNN: A lightweight convolutional neural network for pose-independent body condition score estimation in goats. *Journal of Agriculture and Food Research*, *16*, 101174. <https://doi.org/10.1016/j.jafr.2024.101174>
- Wang, Y., Mücher, S., Wang, W., & Kooistra, L. (2025). Body Weight Estimation of Cattle in Standing and Lying Postures Using Point Clouds Derived from Unmanned Aerial Vehicle-Based LiDAR. *Drones*, *9*(2), 84. <https://doi.org/10.3390/drones9020084>
- Xie, C., Cang, Y., Lou, X., Xiao, H., Xu, X., Li, X., & Zhou, W. (2024). A novel approach based on a modified mask R-CNN for the weight prediction of live pigs. *Artificial Intelligence in Agriculture*, *12*, 19–28. <https://doi.org/10.1016/j.aiia.2024.03.001>
- Xiong, Y., Condotta, I. C. F. S., Musgrave, J. A., Brown-Brandl, T. M., & Mulliniks, J. T. (2023). Estimating body weight and body condition score of mature beef cows using depth images. *Translational Animal Science*, *7*(1). <https://doi.org/10.1093/tas/txad085>
- Xu, B., Mao, Y., Wang, W., & Chen, G. (2024). Intelligent weight prediction of cows based on semantic segmentation and back propagation neural network. *Frontiers in Artificial Intelligence*, *7*. <https://doi.org/10.3389/frai.2024.1299169>
- Zhang, H., Zhang, Y., Niu, K., & He, Z. (2025). Neural network-based method for contactless estimation of carcass weight from live beef images. *Computers and Electronics in Agriculture*, *229*, 109830. <https://doi.org/10.1016/j.compag.2024.109830>

- Zhang, J., Zhuang, Y., Ji, H., & Teng, G. (2021). Pig Weight and Body Size Estimation Using a Multiple Output Regression Convolutional Neural Network: A Fast and Fully Automatic Method. *Sensors*, 21(9), 3218. <https://doi.org/10.3390/s21093218>
- Zhao, K., Zhang, M., Shen, W., Liu, X., Ji, J., Dai, B., & Zhang, R. (2023). Automatic body condition scoring for dairy cows based on efficient net and convex hull features of point clouds. *Computers and Electronics in Agriculture*, 205, 107588. <https://doi.org/10.1016/j.compag.2022.107588>

ANEXOS

Anexo 1: Autorización de acceso a fundo

"Año de la Recuperación y Consolidación de la Economía Peruana"

Zapatero, 20/05/2025

AUTORIZACIÓN DE ACCESO A FUNDO "GANADERÍA DR"

Yo, **DANY CRISTIAN ROMERO LOZANO**, identificado con DNI N.º 70749430, propietario del fundo "GANADERÍA DR", ubicado en el distrito de Zapatero, provincia de Lamas, por medio del presente documento **AUTORIZO** al señor **HECTOR LUIS FERNANDEZ ROSILLO**, identificado con DNI N.º 75166773, estudiante de la Escuela Profesional de Ingeniería de Sistemas e Informática de la Universidad Nacional de San Martín, a ingresar a mi propiedad con la finalidad de tomar fotografías del ganado vacuno, las cuales serán capturadas durante el proceso de pesaje en balanza, en un entorno controlado.

Esta actividad forma parte del desarrollo de su proyecto de tesis titulado: "**Pesaje no invasivo de ganado vacuno mediante modelos híbridos basado en segmentación de imágenes y regresión**", con fines estrictamente académicos y científicos.

Declaro que he sido debidamente informado sobre el propósito de esta recolección de imágenes y que doy mi consentimiento libre y voluntario.

Sin otro particular, suscribo la presente autorización:

DANY CRISTIAN ROMERO LOZANO
DNI N° 70749430

Anexo 2: Código de desarrollo de modelos desarrollados

SE INSTALARON LAS DEPENDENCIAS REQUERIDAS

```
!pip install ultralytics xgboost opencv-python scikit-learn
matplotlib
```

IMPORTAMOS LIBRERÍAS NECESARIAS

```
from PIL import Image
import os
import time
from ultralytics import YOLO
import cv2
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split,
RandomizedSearchCV, KFold
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error,
mean_absolute_error, r2_score
import joblib
import pandas as pd
import lightgbm as lgb
from xgboost import XGBRegressor
from sklearn.model_selection import learning_curve
```

PREPROCESAR (AJUSTAR IMÁGENES RESPECTO AL OBJETO DE REFERENCIA: CÍRCULO)

```
# Variables para reporte
total_images = 0
rotated_images = 0
saved_images = 0
failed_images = 0

# Función para rotar si la imagen está vertical
def rotate_image_if_vertical(image_path):
    global rotated_images
    img = cv2.imread(image_path)
    if img.shape[0] > img.shape[1]:
        print(f"🌀 Rotando imagen vertical: {image_path}")
        img = cv2.rotate(img, cv2.ROTATE_90_COUNTERCLOCKWISE)
        cv2.imwrite(image_path, img)
        rotated_images += 1
    return img

# Cargar modelo YOLO (clase 'object')
```

```

model = YOLO('lastcirculo.pt') # 🔍 Clase segmentada = 'object'

# Rutas
input_folder = '/content/salida900x800'
output_folder = 'salida_estandarizada_obj_referencia'
os.makedirs(output_folder, exist_ok=True)

# Parámetros
target_diameter_px = 30
canvas_size = (900, 800)
mask_color = (0, 255, 0) # Verde

def segment_and_resize(image_path, output_path):
    global saved_images, failed_images

    rotate_image_if_vertical(image_path)

    results = model(image_path, imgsz=1280)

    if not results[0].masks or len(results[0].masks.xy) == 0:
        print(f"❌ No se detectó círculo en: {image_path}")
        failed_images += 1
        return

    mask_coords = results[0].masks.xy[0]
    mask = np.array(mask_coords, dtype=np.int32)

    x_min = np.min(mask[:, 0])
    x_max = np.max(mask[:, 0])
    diameter_detected = x_max - x_min

    if diameter_detected <= 0:
        print(f"❌ Diámetro inválido: {diameter_detected}")
        failed_images += 1
        return

    print(f"📏 Diámetro detectado: {diameter_detected:.2f}px")
    scale_factor = target_diameter_px / diameter_detected
    print(f"🔧 Escala aplicada: {scale_factor:.3f}")

    original_image = cv2.imread(image_path)
    new_size = (int(original_image.shape[1] * scale_factor),
int(original_image.shape[0] * scale_factor))
    resized_image = cv2.resize(original_image, new_size)

    # Escalar la máscara y dibujarla sobre la imagen
    scaled_mask = (mask * scale_factor).astype(np.int32)

```

```

    # Crear una imagen binaria de la máscara escalada
    mask_img = np.zeros((resized_image.shape[0],
resized_image.shape[1]), dtype=np.uint8)
    cv2.fillPoly(mask_img, [scaled_mask], 255)

    # Aplicar suavizado morfológico (quita picos y bordes
irregulares)
    kernel = np.ones((3, 3), np.uint8)
    smoothed = cv2.morphologyEx(mask_img, cv2.MORPH_OPEN, kernel)

    # Extraer contornos
    contours, _ = cv2.findContours(smoothed, cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)

    if len(contours) > 0:
        # Usar el contorno más grande
        largest_contour = max(contours, key=cv2.contourArea)

        # Ajustar un círculo perfecto
        # Usar el centroide del contorno como centro
        M = cv2.moments(largest_contour)
        if M["m00"] != 0:
            cx = int(M["m10"] / M["m00"])
            cy = int(M["m01"] / M["m00"])
            center = (cx, cy)
        else:
            center = (int(x), int(y)) # fallback

        # Forzar un radio exacto
        radius = target_diameter_px // 2

        # Pintar el círculo exacto, centrado
        cv2.circle(resized_image, center, radius, mask_color,
thickness=3)

    # Crear canvas blanco
    canvas = np.ones((canvas_size[1], canvas_size[0], 3),
dtype=np.uint8) * 255

    if new_size[0] > canvas_size[0] or new_size[1] >
canvas_size[1]:
        print(f"⚠ Imagen escalada excede canvas: {new_size} >
{canvas_size}")
        failed_images += 1
        return

    x_offset = (canvas_size[0] - new_size[0]) // 2
    y_offset = (canvas_size[1] - new_size[1]) // 2

```

```

    canvas[y_offset:y_offset + new_size[1], x_offset:x_offset +
new_size[0]] = resized_image

    cv2.imwrite(output_path, canvas)
    print(f"✅ Guardado: {output_path}")
    saved_images += 1

# Procesar todas las imágenes
for filename in os.listdir(input_folder):
    if filename.lower().endswith(('.jpg', '.jpeg', '.png')):
        total_images += 1
        input_path = os.path.join(input_folder, filename)
        output_path = os.path.join(output_folder, filename)
        segment_and_resize(input_path, output_path)

# 🏁 REPORTE FINAL
print("\n📊 REPORTE FINAL")
print("=====")
print(f"📁 Total de imágenes procesadas: {total_images}")
print(f"🔄 Imágenes rotadas (verticales): {rotated_images}")
print(f"✅ Imágenes guardadas correctamente: {saved_images}")
print(f"❌ Imágenes con error o sin círculo: {failed_images}")
print("=====")

```

SEGMENTAR IMÁGENES: VACA ROJA Y FONDO BLANCO

```

# Configuración de rutas
input_folder = '/content/salida_estandarizada_obj_referencia'
output_folder = 'vacas_segmentadas'
os.makedirs(output_folder, exist_ok=True)

# Multiplicador
MULTIPLICADOR = 2

# Modelo YOLOv8 (segmentación de vacas)
model = YOLO('yolo8x-seg.pt')
COW_CLASS_ID = 19
COW_COLOR = (0, 0, 255) # Rojo en BGR

# Resolución objetivo para guardar imágenes segmentadas
RESOLUCION_OBJETIVO = (256, 256)

# Procesar imágenes
for filename in os.listdir(input_folder):
    if filename.lower().endswith(('.jpg', '.jpeg', '.png')):
        input_path = os.path.join(input_folder, filename)
        name, ext = os.path.splitext(filename)

        image = cv2.imread(input_path)

```

```

results = model(input_path, imgsz=640)[0]

output_img = np.ones_like(image) * 255
cow_detected = False

if results.masks is not None and len(results.bboxes.cls) >
0:
    for i, cls in enumerate(results.bboxes.cls):
        if int(cls) == COW_CLASS_ID:
            cow_detected = True
            mask = results.masks.data[i].cpu().numpy()
            mask = (mask * 255).astype(np.uint8)
            mask = cv2.resize(mask, (image.shape[1],
image.shape[0]))

            cow_area = np.zeros_like(image)
            cow_area[:] = COW_COLOR
            cow_mask = cv2.bitwise_and(cow_area, cow_area,
mask=mask)

            output_img = cv2.bitwise_and(output_img,
output_img, mask=cv2.bitwise_not(mask))
            output_img += cow_mask

        if cow_detected:
            # Redimensionar imagen segmentada
            output_img_resized = cv2.resize(output_img,
RESOLUCION_OBJETIVO)

            for i in range(1, MULTIPLICADOR + 1):
                output_filename = f"{i}_{name}.jpg"
                output_path = os.path.join(output_folder,
output_filename)
                cv2.imwrite(output_path, output_img_resized)

                print(f"✅ Segmentado y guardado {MULTIPLICADOR}x:
{filename}")
            else:
                print(f"⚠️ No se detectó vaca en: {filename}")
        else:
            print(f"⚠️ No se detectó vaca en: {filename}")

print("\n🔗 Segmentación y multiplicación completadas.")

```

EXTRACCIÓN DE CARACTERÍSTICAS

```

# 2. Ruta a tu dataset segmentado
ruta_imagenes = '/content/train' # Ajusta si es necesario

```

```

# 3. Función para extraer el peso desde el nombre del archivo
def extraer_peso(nombre_archivo):
    partes = nombre_archivo.split('_')
    try:
        peso = int(partes[4])
        return peso
    except:
        return None

# 4. Función para extraer características geométricas
def extraer_caracteristicas_geometricas(imagen):
    gris = cv2.cvtColor(imagen, cv2.COLOR_BGR2GRAY)
    _, binaria = cv2.threshold(gris, 250, 255,
cv2.THRESH_BINARY_INV)
    contornos, _ = cv2.findContours(binaria, cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)
    if contornos:
        contorno_principal = max(contornos, key=cv2.contourArea)
        area = cv2.contourArea(contorno_principal)
        perimetro = cv2.arcLength(contorno_principal, True)
        x, y, ancho, alto = cv2.boundingRect(contorno_principal)
        relacion_aspecto = ancho / alto if alto != 0 else 0
        momentos_hu =
cv2.HuMoments(cv2.moments(contorno_principal)).flatten()[ :3]
        caracteristicas = [area, perimetro, ancho, alto,
relacion_aspecto] + momentos_hu.tolist()
        return np.array(caracteristicas), contorno_principal, (x,
y, ancho, alto)
    else:
        return np.zeros(8), None, (0,0,0,0)

# 5. Cargar imágenes y preparar dataset
X, y = [], []
nombres_imagenes = []
contornos_lista = []
bounding_boxes = []

for archivo in os.listdir(ruta_imagenes):
    if archivo.endswith('.jpg') or archivo.endswith('.png'):
        peso = extraer_peso(archivo)
        if peso is not None:
            ruta_imagen = os.path.join(ruta_imagenes, archivo)
            imagen = cv2.imread(ruta_imagen)
            if imagen is not None:
                caracteristicas, contorno, bbox =
extraer_caracteristicas_geometricas(imagen)
                X.append(caracteristicas)
                y.append(peso)

```

```

        nombres_imagenes.append(archivo)
        contornos_lista.append(contorno)
        bounding_boxes.append(bbox)

X = np.array(X)
y = np.array(y)
print(f"✅ Cantidad de imágenes procesadas: {len(X)}")

```

GUARDAMOS EL DATASET GEOMÉTRICO Y SU PESO EN UN ARCHIVO .PKL

```

import joblib

# Ruta donde se guardarán los datos
ruta_datos = '/content/dataset_geometrico_y_peso.pkl'

# Guardar X y y en un solo archivo
joblib.dump({'X': X, 'y': y}, ruta_datos)

print(f"✅ Dataset guardado como: {ruta_datos}")

```

ENTRENAMIENTO DEL PRIMER MODELO: RANDOM FOREST

```

import joblib
# Cargar los datos previamente guardados
datos = joblib.load('/content/dataset_geometrico_y_peso.pkl')

# Asignar a variables X e y
X = datos['X']
y = datos['y']

print(f"✅ Dataset cargado. Total de muestras: {len(X)}, Número de
características: {X.shape[1]}")

# 6. Validación cruzada KFold con k=16
cv_kfold_rf = KFold(n_splits=5, shuffle=True, random_state=42)

# 7. Espacio de búsqueda de hiperparámetros
param_rf = {
    'n_estimators': [200, 400, 600, 800], # Aumentamos el máximo y
el mínimo
    'max_depth': [10, 15, 20, 25, None], # Rangos más profundos
    'min_samples_split': [2, 5, 10, 15],
    'min_samples_leaf': [1, 2, 4, 8],
    'max_features': [0.5, 0.7, 0.9, 1.0, 'sqrt', 'log2'],
    'bootstrap': [True]
}

# 8. RandomizedSearchCV con KFold
modelo_base_rf = RandomForestRegressor(random_state=42, n_jobs=-1)

```

```

random_search_rf = RandomizedSearchCV(
    estimator=modelo_base_rf,
    param_distributions=param_rf,
    n_iter=20,
    scoring='neg_mean_squared_error',
    cv=cv_kfold_rf,
    verbose=2,
    random_state=42,
    n_jobs=-1
)

random_search_rf.fit(X, y)

# 9. Métricas por fold
resultados_rf = pd.DataFrame(random_search_rf.cv_results_)
mejor_idx_rf = random_search_rf.best_index_
mejor_fila_rf = resultados_rf.loc[mejor_idx_rf]

folds_mse_rf = []
for i in range(cv_kfold_rf.get_n_splits()):
    key = f'split{i}_test_score'
    if key in resultados_rf.columns:
        folds_mse_rf.append(-mejor_fila_rf[key])

df_metricas_rf = pd.DataFrame({
    'Fold': list(range(1, len(folds_mse_rf)+1)),
    'MSE': folds_mse_rf,
    'RMSE': [np.sqrt(mse) for mse in folds_mse_rf],
    'MAE': [mean_absolute_error(y[test_idx],
random_search_rf.best_estimator_.predict(X[test_idx])) for _,
test_idx in cv_kfold_rf.split(X)],
    'R2': [r2_score(y[test_idx],
random_search_rf.best_estimator_.predict(X[test_idx])) for _,
test_idx in cv_kfold_rf.split(X)]
})

ruta_csv_rf = '/content/metricas_random_forest_kfold16.csv'
df_metricas_rf.to_csv(ruta_csv_rf, index=False)

print("\n📄 Métricas por fold (Random Forest):")
print(df_metricas_rf)
print(f"\n✅ Métricas guardadas en: {ruta_csv_rf}")

# Cargamos el dataset para evaluar
datos = joblib.load('/content/dataset_geometrico_y_peso_test.pkl')

# Asignar a variables X e y
X = datos['X']

```

```

y = datos['y']

print(f"✅ Dataset cargado. Total de muestras: {len(X)}, Número de
características: {X.shape[1]}")

# 10. Evaluación global
mejor_modelo_rf = random_search_rf.best_estimator_
y_pred_rf = mejor_modelo_rf.predict(X)

mse_rf = mean_squared_error(y, y_pred_rf)
rmse_rf = np.sqrt(mse_rf)
mae_rf = mean_absolute_error(y, y_pred_rf)
r2_rf = r2_score(y, y_pred_rf)

print("\n✅ Rendimiento del Mejor Modelo Random Forest (k=5):")
print(f"• MSE   : {mse_rf:.2f}")
print(f"• RMSE  : {rmse_rf:.2f}")
print(f"• MAE   : {mae_rf:.2f}")
print(f"• R2    : {r2_rf:.4f}")
print(f"• Hiperparámetros óptimos:
{random_search_rf.best_params_}")

joblib.dump(mejor_modelo_rf,
            '/content/mejor_modelo_random_forest_kfold16.pkl')
print("\n✅ Modelo guardado como
'mejor_modelo_random_forest_kfold16.pkl'")

```

GRÁFICAS DEL MODELO RANDOM FOREST

```

# 10. Visualizar importancia de características
feature_names = ['Área', 'Perímetro', 'Ancho', 'Alto', 'Relación
aspecto', 'HuMoments1', 'HuMoments2', 'HuMoments3']
importancias = mejor_modelo_rf.feature_importances_

plt.figure(figsize=(10, 6))
sns.barplot(x=importancias, y=feature_names)
plt.title('Importancia de características geométricas')
plt.xlabel('Importancia')
plt.ylabel('Característica')
plt.grid(True)
plt.show()

# 3. Distribución de errores (residuales)
errores = y - y_pred_rf
plt.figure(figsize=(8, 6))
sns.histplot(errores, kde=True, bins=30, color='orange')
plt.title('Distribución de errores (residuales)')
plt.xlabel('Error (real - predicho)')
plt.grid(True)

```

```

plt.show()

# 4. Curva de aprendizaje
from sklearn.model_selection import learning_curve

train_sizes, train_scores, test_scores = learning_curve(
    mejor_modelo_rf, X, y, cv=5, scoring='r2', n_jobs=-1,
    train_sizes=np.linspace(0.1, 1.0, 10)
)
train_scores_mean = np.mean(train_scores, axis=1)
test_scores_mean = np.mean(test_scores, axis=1)

plt.figure(figsize=(10, 6))
plt.plot(train_sizes, train_scores_mean, 'o-',
label='Entrenamiento')
plt.plot(train_sizes, test_scores_mean, 'o-', label='Validación')
plt.title('Curva de aprendizaje (Random Forest)')
plt.xlabel('Tamaño del conjunto de entrenamiento')
plt.ylabel('Score R2')
plt.legend()
plt.grid(True)
plt.show()

# 5. Gráfico de errores residuales
plt.figure(figsize=(8, 6))
sns.scatterplot(x=y_pred_rf, y=errores)
plt.axhline(y=0, color='r', linestyle='--')
plt.title('Gráfico de errores residuales')
plt.xlabel('Predicción')
plt.ylabel('Error (real - predicho)')
plt.grid(True)
plt.show()

# 11. Visualizar diagrama de dispersión Predicción vs Real
plt.figure(figsize=(8, 8))
sns.scatterplot(x=y, y=y_pred_rf)
plt.plot([y.min(), y.max()], [y.min(), y.max()], 'r--') # Línea
ideal
plt.title('Predicción vs Valor Real (Random Forest)')
plt.xlabel('Peso real (kg)')
plt.ylabel('Peso predicho (kg)')
plt.grid(True)
plt.show()

# 12. Visualizar características gráficamente de la primera imagen
primera_imagen_path = os.path.join(ruta_imagenes,
nombres_imagenes[0])
primera_imagen = cv2.imread(primera_imagen_path)

```

```

primera_imagen_rgb = cv2.cvtColor(primera_imagen,
cv2.COLOR_BGR2RGB)
contorno = contornos_lista[0]
(x, y, ancho, alto) = bounding_boxes[0]

# Área rellena
imagen_area = primera_imagen_rgb.copy()
cv2.drawContours(imagen_area, [contorno], -1, (0,255,0),
thickness=cv2.FILLED)

# Perímetro
imagen_perimetro = primera_imagen_rgb.copy()
cv2.drawContours(imagen_perimetro, [contorno], -1, (0,255,0),
thickness=2)

# Bounding box
imagen_bounding = primera_imagen_rgb.copy()
cv2.rectangle(imagen_bounding, (x,y), (x+ancho, y+alto),
(255,0,255), 2)

# Mostrar imágenes
fig, axs = plt.subplots(1, 3, figsize=(18, 6))

axs[0].imshow(imagen_area)
axs[0].set_title(f'Área Rellena (verde)')
axs[0].axis('off')

axs[1].imshow(imagen_perimetro)
axs[1].set_title(f'Perímetro Dibujado (azul)')
axs[1].axis('off')

axs[2].imshow(imagen_bounding)
axs[2].set_title(f'Bounding Box (morado)')
axs[2].axis('off')

plt.suptitle(f"Visualización de características -
{nombres_imagenes[0]}", fontsize=16)
plt.show()

# 13. Imprimir características geométricas de la primera imagen
caracteristicas_extraidas, _, _ =
extraer_caracteristicas_geometricas(cv2.imread(primera_imagen_path)
)

feature_names = ['Área', 'Perímetro', 'Ancho', 'Alto', 'Relación
aspecto', 'HuMoments1', 'HuMoments2', 'HuMoments3']

print("\n📄 Características geométricas de la primera imagen:")

```

```
for nombre, valor in zip(feature_names, características_extraidas):
    print(f"• {nombre}: {valor:.2f}")
```

ENTRENAMIENTO DEL SEGUNDO MODELO: XGBOOST

```
import joblib

# Cargar los datos previamente guardados
datos = joblib.load('/content/dataset_geometrico_y_peso.pkl')

# Asignar a variables X e y
X = datos['X']
y = datos['y']

print(f"✓ Dataset cargado. Total de muestras: {len(X)}, Número de
características: {X.shape[1]}")

# 6. Validación cruzada KFold
cv_kfold_xgb = KFold(n_splits=5, shuffle=True, random_state=42)

# 7. Espacio de hiperparámetros
param_xgb = {
    'n_estimators': [100, 200, 300],
    'learning_rate': [0.03, 0.05, 0.1],
    'max_depth': [3, 4, 5, 6],
    'subsample': [0.7, 0.8, 0.9],
    'colsample_bytree': [0.7, 0.8, 0.9],
    'gamma': [0, 0.1, 0.5, 1.0],
    'reg_alpha': [0, 0.005, 0.01, 0.1],
    'reg_lambda': [1, 1.5, 2]
}

# 8. Búsqueda de hiperparámetros
modelo_base_xgb = XGBRegressor(random_state=42, n_jobs=-1)
random_search_xgb = RandomizedSearchCV(
    estimator=modelo_base_xgb,
    param_distributions=param_xgb,
    n_iter=20,
    scoring='neg_mean_squared_error',
    cv=cv_kfold_xgb,
    verbose=2,
    random_state=42,
    n_jobs=-1
)

random_search_xgb.fit(X, y)

# 9. Métricas por fold
resultados_xgb = pd.DataFrame(random_search_xgb.cv_results_)
```

```

mejor_idx_xgb = random_search_xgb.best_index_
mejor_fila_xgb = resultados_xgb.loc[mejor_idx_xgb]

folds_mse_xgb = []
for i in range(cv_kfold_xgb.get_n_splits()):
    key = f'split{i}_test_score'
    if key in resultados_xgb.columns:
        folds_mse_xgb.append(-mejor_fila_xgb[key])

df_metricas_xgb = pd.DataFrame({
    'Fold': list(range(1, len(folds_mse_xgb)+1)),
    'MSE': folds_mse_xgb,
    'RMSE': [np.sqrt(mse) for mse in folds_mse_xgb],
    'MAE': [mean_absolute_error(y[test_idx],
random_search_xgb.best_estimator_.predict(X[test_idx])) for _,
test_idx in cv_kfold_xgb.split(X)],
    'R2': [r2_score(y[test_idx],
random_search_xgb.best_estimator_.predict(X[test_idx])) for _,
test_idx in cv_kfold_xgb.split(X)]
})

ruta_csv_xgb = '/content/metricas_xgboost_kfold16.csv'
df_metricas_xgb.to_csv(ruta_csv_xgb, index=False)

print("\n📄 Métricas por fold (XGBoost):")
print(df_metricas_xgb)
print(f"\n✅ Métricas guardadas en: {ruta_csv_xgb}")

# Cargamos el dataset para evaluar
datos = joblib.load('/content/dataset_geometrico_y_peso_test.pkl')

# Asignar a variables X e y
X = datos['X']
y = datos['y']

print(f"✅ Dataset cargado. Total de muestras: {len(X)}, Número de
características: {X.shape[1]}")

# 10. Evaluación global
mejor_modelo_xgb = random_search_xgb.best_estimator_
y_pred_xgb = mejor_modelo_xgb.predict(X)

mse_xgb = mean_squared_error(y, y_pred_xgb)
rmse_xgb = np.sqrt(mse_xgb)
mae_xgb = mean_absolute_error(y, y_pred_xgb)
r2_xgb = r2_score(y, y_pred_xgb)

print("\n✅ Rendimiento del Mejor Modelo XGBoost (k=5):")

```

```

print(f"• MSE   : {mse_xgb:.2f}")
print(f"• RMSE  : {rmse_xgb:.2f}")
print(f"• MAE   : {mae_xgb:.2f}")
print(f"• R2    : {r2_xgb:.4f}")
print(f"• Hiperparámetros óptimos:
{random_search_xgb.best_params_}")

joblib.dump(mejor_modelo_xgb,
            '/content/mejor_modelo_xgboost_kfold16.pkl')
print("\n✅ Modelo guardado como
'mejor_modelo_xgboost_kfold16.pkl'")

```

GRÁFICAS DEL MODELO XGBOOST

```

# 10. Importancia de características
feature_names = ['Área', 'Perímetro', 'Ancho', 'Alto', 'Relación
aspecto', 'HuMoments1', 'HuMoments2', 'HuMoments3']
importancias = mejor_modelo_xgb.feature_importances_

plt.figure(figsize=(10, 6))
sns.barplot(x=importancias, y=feature_names)
plt.title('Importancia de características geométricas (XGBoost)')
plt.xlabel('Importancia')
plt.ylabel('Característica')
plt.grid(True)
plt.show()

# 11. Distribución de errores
errores_xgb = y - y_pred_xgb
plt.figure(figsize=(8, 6))
sns.histplot(errores_xgb, kde=True, bins=30, color='orange')
plt.title('Distribución de errores residuales (XGBoost)')
plt.xlabel('Error (real - predicho)')
plt.grid(True)
plt.show()

# 12. Curva de aprendizaje
train_sizes, train_scores, test_scores = learning_curve(
    mejor_modelo_xgb, X, y, cv=5, scoring='r2', n_jobs=-1,
    train_sizes=np.linspace(0.1, 1.0, 10)
)
train_scores_mean = np.mean(train_scores, axis=1)
test_scores_mean = np.mean(test_scores, axis=1)

plt.figure(figsize=(10, 6))
plt.plot(train_sizes, train_scores_mean, 'o-',
label='Entrenamiento')
plt.plot(train_sizes, test_scores_mean, 'o-', label='Validación')
plt.title('Curva de aprendizaje (XGBoost)')

```

```

plt.xlabel('Tamaño del conjunto de entrenamiento')
plt.ylabel('Score R2')
plt.legend()
plt.grid(True)
plt.show()

# 13. Gráfico de errores residuales
plt.figure(figsize=(8, 6))
sns.scatterplot(x=y_pred_xgb, y=errores_xgb)
plt.axhline(y=0, color='r', linestyle='--')
plt.title('Errores residuales (XGBoost)')
plt.xlabel('Predicción')
plt.ylabel('Error (real - predicho)')
plt.grid(True)
plt.show()

# 14. Diagrama Predicción vs Real
plt.figure(figsize=(8, 8))
sns.scatterplot(x=y, y=y_pred_xgb)
plt.plot([y.min(), y.max()], [y.min(), y.max()], 'r--')
plt.title('Predicción vs Valor Real (XGBoost)')
plt.xlabel('Peso real (kg)')
plt.ylabel('Peso predicho (kg)')
plt.grid(True)
plt.show()

```

ENTRENAMIENTO DEL TERCER MODELO: LIGHT GRADIENT BOOSTING MACHINE (LIGHTGBM)

```

import joblib

# Cargar los datos previamente guardados
datos = joblib.load('/content/dataset_geometrico_y_peso.pkl')

# Asignar a variables X e y
X = datos['X']
y = datos['y']

print(f"✅ Dataset cargado. Total de muestras: {len(X)}, Número de características: {X.shape[1]}")

# 6. Validación cruzada KFold
cv_kfold_lgb = KFold(n_splits=5, shuffle=True, random_state=42)

# 7. Espacio de hiperparámetros
param_lgb = {
    'n_estimators': [300, 400, 500],
    'learning_rate': [0.03, 0.05, 0.1],

```

```

    'num_leaves': [20, 31, 40, 50, 60],
    'max_depth': [-1, 5, 7, 10],
    'min_child_samples': [20, 30, 40],
    'subsample': [0.7, 0.8, 0.9],
    'colsample_bytree': [0.7, 0.8, 0.9],
    'reg_alpha': [0, 0.005, 0.01, 0.1],
    'reg_lambda': [1, 1.5, 2]
}

# 8. RandomizedSearchCV con KFold
modelo_base_lgb = lgb.LGBMRegressor(random_state=42, n_jobs=-1)
random_search_lgb = RandomizedSearchCV(
    estimator=modelo_base_lgb,
    param_distributions=param_lgb,
    n_iter=20,
    scoring='neg_mean_squared_error',
    cv=cv_kfold_lgb,
    verbose=2,
    random_state=42,
    n_jobs=-1
)

random_search_lgb.fit(X, y)

# 9. Métricas por fold
resultados_lgb = pd.DataFrame(random_search_lgb.cv_results_)
mejor_idx_lgb = random_search_lgb.best_index_
mejor_fila_lgb = resultados_lgb.loc[mejor_idx_lgb]

folds_mse_lgb = []
for i in range(cv_kfold_lgb.get_n_splits()):
    key = f'split{i}_test_score'
    if key in resultados_lgb.columns:
        folds_mse_lgb.append(-mejor_fila_lgb[key])

df_metricas_lgb = pd.DataFrame({
    'Fold': list(range(1, len(folds_mse_lgb)+1)),
    'MSE': folds_mse_lgb,
    'RMSE': [np.sqrt(mse) for mse in folds_mse_lgb],
    'MAE': [mean_absolute_error(y[test_idx],
random_search_lgb.best_estimator_.predict(X[test_idx])) for _,
test_idx in cv_kfold_lgb.split(X)],
    'R2': [r2_score(y[test_idx],
random_search_lgb.best_estimator_.predict(X[test_idx])) for _,
test_idx in cv_kfold_lgb.split(X)]
})

ruta_csv_lgb = '/content/metricas_lightgbm_kfold16.csv'

```

```

df_metricas_lgb.to_csv(ruta_csv_lgb, index=False)

print("\n📄 Métricas por fold (LightGBM):")
print(df_metricas_lgb)
print(f"\n✅ Métricas guardadas en: {ruta_csv_lgb}")

# Cargamos el dataset para evaluar
datos = joblib.load('/content/dataset_geometrico_y_peso_test.pkl')

# Asignar a variables X e y
X = datos['X']
y = datos['y']

print(f"✅ Dataset cargado. Total de muestras: {len(X)}, Número de
características: {X.shape[1]}")

# 10. Evaluación global
mejor_modelo_lgb = random_search_lgb.best_estimator_
y_pred_lgb = mejor_modelo_lgb.predict(X)

mse_lgb = mean_squared_error(y, y_pred_lgb)
rmse_lgb = np.sqrt(mse_lgb)
mae_lgb = mean_absolute_error(y, y_pred_lgb)
r2_lgb = r2_score(y, y_pred_lgb)

print("\n✅ Rendimiento del Mejor Modelo LightGBM (k=5):")
print(f"• MSE   : {mse_lgb:.2f}")
print(f"• RMSE  : {rmse_lgb:.2f}")
print(f"• MAE   : {mae_lgb:.2f}")
print(f"• R2    : {r2_lgb:.4f}")
print(f"• Hiperparámetros óptimos:
{random_search_lgb.best_params}")

joblib.dump(mejor_modelo_lgb,
'/content/mejor_modelo_lightgbm_kfold16.pkl')
print("\n✅ Modelo guardado como
'mejor_modelo_lightgbm_kfold16.pkl'")

```

GRÁFICAS DEL MODELO LIGHTGBM

```

# 10. Importancia de características
feature_names = ['Área', 'Perímetro', 'Ancho', 'Alto', 'Relación
aspecto', 'HuMoments1', 'HuMoments2', 'HuMoments3']
importancias = mejor_modelo_lgb.feature_importances_

plt.figure(figsize=(10, 6))
sns.barplot(x=importancias, y=feature_names)
plt.title('Importancia de características geométricas (LightGBM)')
plt.xlabel('Importancia')

```

```

plt.ylabel('Característica')
plt.grid(True)
plt.show()

# 11. Distribución de errores
errores_lgb = y - y_pred_lgb
plt.figure(figsize=(8, 6))
sns.histplot(errores_lgb, kde=True, bins=30, color='orange')
plt.title('Distribución de errores residuales (LightGBM)')
plt.xlabel('Error (real - predicho)')
plt.grid(True)
plt.show()

# 12. Curva de aprendizaje
train_sizes, train_scores, test_scores = learning_curve(
    mejor_modelo_lgb, X, y, cv=5, scoring='r2', n_jobs=-1,
    train_sizes=np.linspace(0.1, 1.0, 10)
)
train_scores_mean = np.mean(train_scores, axis=1)
test_scores_mean = np.mean(test_scores, axis=1)

plt.figure(figsize=(10, 6))
plt.plot(train_sizes, train_scores_mean, 'o-',
label='Entrenamiento')
plt.plot(train_sizes, test_scores_mean, 'o-', label='Validación')
plt.title('Curva de aprendizaje (LightGBM)')
plt.xlabel('Tamaño del conjunto de entrenamiento')
plt.ylabel('Score R²')
plt.legend()
plt.grid(True)
plt.show()

# 13. Gráfico de errores residuales
plt.figure(figsize=(8, 6))
sns.scatterplot(x=y_pred_lgb, y=errores_lgb)
plt.axhline(y=0, color='r', linestyle='--')
plt.title('Errores residuales (LightGBM)')
plt.xlabel('Predicción')
plt.ylabel('Error (real - predicho)')
plt.grid(True)
plt.show()

# 14. Diagrama Predicción vs Real
plt.figure(figsize=(8, 8))
sns.scatterplot(x=y, y=y_pred_lgb)
plt.plot([y.min(), y.max()], [y.min(), y.max()], 'r--')
plt.title('Predicción vs Valor Real (LightGBM)')
plt.xlabel('Peso real (kg)')

```

```
plt.ylabel('Peso predicho (kg)')
plt.grid(True)
plt.show()
```

SEGMENTACIÓN DEL DATASET EN 16 SUBCONJUNTOS DE PRUEBA

```
import joblib
import numpy as np
from sklearn.model_selection import KFold
import os

# Cargar el dataset original
datos = joblib.load('/content/dataset_geometrico_y_peso_test.pkl')
X = datos['X']
y = datos['y']

# Número de folds
K = 16

# Crear carpeta para guardar corridas
os.makedirs('/content/corridas_k16', exist_ok=True)

# Generador de particiones
kf = KFold(n_splits=K, shuffle=False)

# Convertimos a índices directamente para control manual
indices = np.arange(len(X))
folds = list(kf.split(indices)) # Genera K tuplas (train_idx,
val_idx)

# Crear los 16 archivos
for i in range(K):
    _, val_idx = folds[i] # solo necesitamos los índices de la
parte que se excluye
    mask = np.ones(len(X), dtype=bool)
    mask[val_idx] = False # excluimos la parte actual

    X_corrida = X[mask]
    y_corrida = y[mask]

    # Guardar
    ruta_salida = f'/content/corridas_k16/corrida_{i+1}.pkl'
    joblib.dump({'X': X_corrida, 'y': y_corrida}, ruta_salida)
    print(f"✅ Archivo generado: corrida_{i+1}.pkl →
{X_corrida.shape[0]} muestras")

print("\n📁 ;Todas las corridas fueron generadas exitosamente!")
```

EVALUACIÓN DEL DESEMPEÑO DE LOS MODELOS EN LOS 16 SUBCONJUNTOS

```

import joblib
import os
import numpy as np
import pandas as pd
from sklearn.metrics import r2_score, mean_absolute_error,
mean_squared_error

# Cargar modelos previamente entrenados
modelo_rf =
joblib.load('/content/mejor_modelo_random_forest_kfold16.pkl')
modelo_xgb =
joblib.load('/content/mejor_modelo_xgboost_kfold16.pkl')
modelo_lgb =
joblib.load('/content/mejor_modelo_lightgbm_kfold16.pkl')

# Ruta de las corridas
ruta_corridas = '/content/corridas_k16'

# Inicializar lista de resultados
resultados = []

# Evaluar las 16 corridas
for i in range(1, 17):
    # Cargar dataset de corrida i
    datos = joblib.load(os.path.join(ruta_corridas,
f'corrida_{i}.pkl'))
    X_corrida = datos['X']
    y_corrida = datos['y']

    # Predicciones
    y_rf = modelo_rf.predict(X_corrida)
    y_xgb = modelo_xgb.predict(X_corrida)
    y_lgb = modelo_lgb.predict(X_corrida)

    # Métricas
    fila = {
        'Corrida': f'corrida_{i}.pkl',
        # Random Forest
        'RF_R2': r2_score(y_corrida, y_rf),
        'RF_MAE': mean_absolute_error(y_corrida, y_rf),
        'RF_RMSE': np.sqrt(mean_squared_error(y_corrida, y_rf)),
        # XGBoost
        'XGB_R2': r2_score(y_corrida, y_xgb),
        'XGB_MAE': mean_absolute_error(y_corrida, y_xgb),
        'XGB_RMSE': np.sqrt(mean_squared_error(y_corrida, y_xgb)),
        # LightGBM
        'LGB_R2': r2_score(y_corrida, y_lgb),
        'LGB_MAE': mean_absolute_error(y_corrida, y_lgb),

```

```

        'LGB_RMSE': np.sqrt(mean_squared_error(y_corrida, y_lgb)),
    }

    resultados.append(fila)

# Crear DataFrame con todas las corridas
df_resultados = pd.DataFrame(resultados)

# Calcular promedios y añadir como fila final
promedios = df_resultados.drop(columns='Corrida').mean().to_dict()
promedios['Corrida'] = 'Promedio'
df_resultados = pd.concat([df_resultados,
pd.DataFrame([promedios])], ignore_index=True)

# Mostrar la tabla
pd.set_option('display.float_format', '{:.4f}'.format)
# Redondear a 4 decimales antes de exportar
df_resultados_redondeado = df_resultados.copy()
columnas_numericas =
df_resultados_redondeado.select_dtypes(include=[np.number]).columns
df_resultados_redondeado[columnas_numericas] =
df_resultados_redondeado[columnas_numericas].round(4)
# Exportar a Excel
df_resultados_redondeado.to_excel('/content/resultados_modelos_corridas.xlsx', index=False)

print("\n📄 Comparación de métricas por modelo en las 16 corridas:")
display(df_resultados_redondeado)

```

Hector Fernandez

Pesaje no invasivo de ganado vacuno mediante modelos híbridos basado en segmentación de imágenes y regresión

 Revisión de Tesis final - Unidad de Investigación FISI

Detalles del documento

Identificador de la entrega

trn:oid::3117:580883964

Fecha de entrega

20 abr 2026, 9:56 GMT-5

Fecha de descarga

20 abr 2026, 10:40 GMT-5

Nombre del archivo

Tesis - Héctor Luis Fernández Rosillo.pdf

Tamaño del archivo

1.5 MB

94 páginas

23.650 palabras

136.412 caracteres




9% Similitud general

El total combinado de todas las coincidencias, incluidas las fuentes superpuestas, para ca...

Filtrado desde el informe

- ▶ Bibliografía
- ▶ Texto citado
- ▶ Texto mencionado
- ▶ Coincidencias menores (menos de 10 palabras)

Fuentes principales

- 7%  Fuentes de Internet
- 2%  Publicaciones
- 7%  Trabajos entregados (trabajos del estudiante)

Marcas de integridad

N.º de alertas de integridad para revisión

Los algoritmos de nuestro sistema analizan un documento en profundidad para buscar inconsistencias que permitirían distinguirlo de una entrega normal. Si advertimos algo extraño, lo marcamos como una alerta para que pueda revisarlo.

Una marca de alerta no es necesariamente un indicador de problemas. Sin embargo, recomendamos que preste atención y la revise.