

Detección de objetos usando aprendizaje profundo aplicado a reconocimiento de ganado bovino

Juan Mauricio Sierra Valencia
Vanessa Alexandra Lopera Mazo

Trabajo de grado presentado para optar al título de
Especialista en Analítica y Ciencia de Datos

Asesor:
Sebastián Rodríguez Colinas, Magíster en Ingeniería

Universidad de Antioquia
Facultad de Ingeniería – Departamento de Ingeniería de Sistemas
Medellín, Colombia
2023

Cita	Sierra Valencia y Lopera Mazo [1]
Referencia Estilo IEEE (2020)	[1] J. M. Sierra Valencia y V. A. Lopera Mazo, “Detección de objetos usando aprendizaje profundo aplicado a reconocimiento de ganado bovino”, Trabajo de grado, Especialización en Analítica y Ciencia de Datos, Universidad de Antioquia, Medellín, 2023.

creative commons CC BY NC SA

Especialización en Analítica y Ciencia de Datos, Cohorte IV.

Centro de Investigaciones Ambientales y de Ingeniería (CIA)

UNIVERSIDAD DE ANTIOQUIA
Vicerrectoría de Docencia

Sistema de Bibliotecas

Centro de Documentación Ingeniería CENDOI

Repositorio Institucional: http://bibliotecadigital.udea.edu.co

Universidad de Antioquia - www.udea.edu.co

Rector: John Jairo Arboleda Céspedes.

Decano/Director: Julio César Saldarriaga Molina

Jefe departamento: Diego José Luis Botia Valderrama.

El contenido de esta obra corresponde al derecho de expresión de los autores y no compromete el pensamiento institucional de la Universidad de Antioquia ni desata su responsabilidad frente a terceros. Los autores asumen la responsabilidad por los derechos de autor y conexos.

Tabla de contenido

RESUMEN 7
ABSTRACT 8
I. INTRODUCCIÓN 9
II. PLANTEAMIENTO DEL PROBLEMA 11
III. OBJETIVOS 12
3.1 Objetivo General 12
3.2 Objetivos Específicos 12
IV. MARCO TEÓRICO 13
4.1 Métricas de evaluación 13
V. DATOS 16
VI. METODOLOGÍA 17
6.1 Preprocesamiento de los datos 17
6.1.1 Selección del Dataset Inicial y Train/Test Split 17
6.1.2 Data Augmentation al Dataset de entrenamiento 21
6.2 Análisis de robustez de los modelos 23
6.3 Iteraciones para la selección del modelo 25
VII. RESULTADOS 27
7.1 Comparación del desempeño de los modelos 27
7.2 Resultados del análisis de robustez de los modelos 29
7.2.1 Análisis de robustez con escenarios que fueron usados en el entrenamiento 29
7.2.1 Análisis de robustez con escenarios que no fueron usados en el entrenamiento 32
VIII. BACKEND y FRONTEND 37
8.1 Backend 37
8.2 Frontend 37
IX. CONCLUSIONES Y RECOMENDACIONES 38
9.1 Conclusiones 38
9.2 Recomendaciones 38

LISTA DE FIGURAS

Figura 1. *Criterios y principios del modelo desarrollado por Welfare Quality para medir el bienestar animal.*

Figura 2. *Ejemplo de algunos elementos que no se etiquetaron.*

Figura 3. *Comparación de los clusters formados después de aplicar los algoritmos Spectral Clustering, KMedoids y Kmeans.*

Figura 4. *Análisis de Silueta que permite identificar que el número óptimo de clusters es 4.*

Figura 5*. muestra de imágenes que fueron incluídas en el Cluster 0.*

Figura 6. *muestra de imágenes que fueron incluídas en el Cluster 1.*

Figura 7. *muestra de imágenes que fueron incluídas en el Cluster 2.*

Figura 8. *muestra de imágenes que fueron incluídas en el Cluster 3.*

Figura 9 . *Ejemplo de algunas transformaciones básicas.*

Figura 10. *muestra de las posibles transformaciones aplicadas al conjunto de datos de entrenamiento.*

Figura 11. *Ejemplo de las transformaciones aplicadas a una imagen en los escenarios Downscale, Fog, Snow, Night y Perspective.*

Figura 12. *Ejemplo de las transformaciones aplicadas a una imagen en los escenarios Rain, Bright, HueSaturation y Shadow (Escenarios incluidos en el entrenamiento).*

Figura 13. *Iteraciones del modelo y despliegue en producción.*

Figura 14. *Matrices de confusión de los modelos evaluados.*

Figura 15. *Métricas de desempeño de los modelos evaluados.*

Figura 16. *Curvas F1-Confidence de los modelos evaluados.*

Figura 17. *Métricas de desempeño de los modelos evaluados con los conjuntos de datos creados para medir la robustez de los mismos.*

Figura 18. *Matrices de confusión de los modelos evaluados con el dataset que simula sombras aleatorias sobre la imagen.*

Figura 19. *Matrices de confusión de los modelos evaluados con el dataset que simula lluvia.*

Figura 20. *Matrices de confusión de los modelos evaluados con el dataset que simula cambios en el color y saturación de la imagen.*

Figura 21. *Matrices de confusión de los modelos evaluados con el dataset que aumenta el brillo en las imágenes.*

Figura 22. *Curvas F1-Confidence de los modelos evaluados ante los escenarios Rain, Bright, HueSaturation y Shadow.*

Figura 23. *Métricas de desempeño de los modelos evaluados con los conjuntos de datos creados para medir la robustez ante los escenarios Downscale, Fog, Snow, Night y Perspective.*

Figura 24. *Matrices de confusión de los modelos evaluados con el dataset que simula condiciones de baja resolución.*

Figura 25. *Matrices de confusión de los modelos evaluados con el dataset que simula condiciones de niebla.*

Figura 26. *Matrices de confusión de los modelos evaluados con el dataset que simula imágenes capturadas en la noche.*

Figura 27. *Matrices de confusión de los modelos evaluados con el dataset que simula cambios en la perspectiva.*

Figura 28. *Matrices de confusión de los modelos evaluados con el dataset que simula condiciones de nieve.*

Figura 29. *Curvas F1-Confidence de los modelos evaluados ante diferentes escenarios*

Figura 30. *Funcionamiento de la aplicación web.*

SIGLAS, ACRÓNIMOS Y ABREVIATURAS

Cms.	Centímetros
Esp.	Especialista
ERIC	Education Resources Information Center
IEEE	Institute of Electrical and Electronics Engineers
MP	Magistrado Ponente
MSc	Magister Scientiae
Párr.	Párrafo
PhD	Philosophiae Doctor
PBQ-SF	Personality Belief Questionnaire Short Form
PostDoc	PostDoctor
UdeA	Universidad de Antioquia

RESUMEN

Para garantizar que cualquier sistema de explotación animal sea sostenible se ha demostrado la importancia de garantizar los niveles óptimos de bienestar animal, donde dicho concepto se entiende como una medida multidimensional que incluye tanto la salud y funcionamiento biológico, el estado afectivo y la comodidad del individuo [1].

El monitoreo del bienestar animal representa un gran reto, ya que es un atributo individual que no puede ser medido directamente, sino que debe evaluarse mediante el uso de indicadores o variables que aporten información al respecto. Dichas evaluación se realizan generalmente mediante inspecciones visuales que, a pesar de que se han desarrollado programas de formación para obtener una alta repetibilidad en los resultados, siguen estando limitados por los largos periodos de tiempo de recopilación de datos y observación que se requieren, lo que implica la contratación de personal dedicado y puede reflejarse en sobrecostos.

Debido a lo anterior, ha crecido la importancia de buscar sistemas automatizados que permitan abordar esta problemática, y, un factor común dentro de dichos sistemas es la aplicación de inteligencia artificial en combinación con la instalación de múltiples sensores que permitan medir las variables indicadoras en cuestión [2],[3]. En el presente trabajo se aborda el monitoreo de ganado vacuno en pastoreo libre mediante la aplicación de algoritmos de detección de objetos que permitan identificar un individuo, sirviendo como primer paso para el desarrollo de los modelos más complejos que permitan medir variables como la cantidad de alimento consumido, identificar comportamientos o posturas asociados a enfermedades o indicadores de estrés.

***Palabras clave* — Bienestar animal, detección de objetos, ganado vacuno.**

ABSTRACT

To ensure that any animal exploitation system is solid and sustainable, the importance of guaranteeing optimal levels of animal welfare has been demonstrated. The concept of animal welfare is understood as a multidimensional measure that includes both the health and biological functioning, the affective state, and the comfort of the individual [1].

The monitoring of animal welfare represents a great challenge since it is an individual attribute that cannot be measured directly but must be evaluated through the use of indicators or variables that provide information in this regard. These evaluations are generally carried out through visual inspections, and although training programs have been developed to obtain highly repeatable results, they are still limited by the long periods of time required for data collection and observation, which implies the hiring of dedicated personnel and can result in additional costs.

Because of this, it has become increasingly important to look for automated systems to address this problem. A common factor within such systems is the application of artificial intelligence in combination with the installation of multiple sensors to measure the indicator variables in question [2],[3]. This paper addresses the monitoring of free-grazing cattle through the application of object detection algorithms to identify an individual, as a first step for the development of more complex models to measure variables such as the amount of food consumed, and to identify behaviors or postures associated with diseases or stress indicators.

***Keywords* — Animal welfare, object detection, cattle.**

I. INTRODUCCIÓN

A lo largo de la historia se ha venido ratificando la importancia de garantizar índices adecuados de bienestar animal (BA) en cualquier sistema de explotación, tanto para mantener y mejorar la calidad y desempeño de los individuos como para alinearse con la creciente necesidad de tener cadenas de producción que reconocen a los animales como seres capaces de experimentar dolor y estrés y que por lo tanto, aseguren una buena alimentación, buenas instalaciones y buena sanidad, que se vean reflejados en el buen comportamiento de los mismos [1].

Welfare Quality® es el proyecto de investigación Europeo, que documenta la incorporación del BA en el proceso de calidad alimentario, incluyendo producción, calidad de transporte y sacrificio, identificando 12 criterios de evaluación que pueden observarse en la Figura 1. Éstos criterios son atributos individuales que no pueden ser medidos directamente, sino que deben evaluarse mediante el uso de indicadores o variables que aporten información al respecto. Dichas evaluaciones se realizan generalmente mediante inspecciones visuales que, a pesar de que se han desarrollado programas de formación para obtener una alta repetibilidad en los resultados, siguen siendo un reto por los largos periodos de tiempo de recopilación de datos y observación que se requieren, lo que implica la contratación de personal dedicado y que puede reflejarse en sobrecostos.

4 PRINCIPIOS	12 CRITERIOS
BUENA ALIMENTACIÓN	1. Ausencia de hambre prolongada
	2. Ausencia de sed prolongada
BUEN ALOJAMIENTO	3. Confort durante el descanso
	4. Confort térmico
	5. Facilidad de movimiento
BUENA SALUD	6. Ausencia de lesiones
	7. Ausencia de enfermedades
	8. Ausencia de dolor inducido por el manejo
COMPORTAMIENTO APROPIADO	9. Expresión de la conducta social
	10. Expresión de otras conductes
	11. Buena relación humano-animal
Welfare Quality	12. Estado emocional positivo

Figura 1. Criterios y principios del modelo desarrollado por Welfare Quality para medir el bienestar animal [4].

Para abordar los retos que representa la medición del BA se han propuesto metodologías que implementan el monitoreo continuo por medio de múltiples sensores con la aplicación de inteligencia artificial que permita procesar la información y convertirla en indicadores útiles a la hora de ser utilizados como indicadores para medir los diferentes criterios que componen el BA [5],[6].

Por su puesto, una de las grandes aplicaciones que tienen estas metodologías se encuentra asociado a la industria ganadera, específicamente para individuos que se encuentran en pastoreo libre lo cual representa retos adicionales para las inspecciones visuales que deben realizarse. Ejemplos de variables que necesitan ser abordadas como indicadores del BA en este caso son, por ejemplo, la cantidad de alimento que consume un animal en un periodo de tiempo, posturas anormales que puedan ser causadas por lesiones o enfermedades, cantidad de descanso que está teniendo un animal, tiempo que tarda rumiando el animal, entre otras.

En el presente trabajo se aborda el monitoreo de ganado vacuno en pastoreo libre mediante la aplicación de algoritmos de detección de objetos que permitan identificar un individuo, sirviendo como primer paso para el desarrollo de los modelos más complejos que permitan medir variables como las antes mencionadas.

II. PLANTEAMIENTO DEL PROBLEMA

El bienestar animal en los sistemas de explotación es un aspecto fundamental para garantizar la calidad y el desempeño de los individuos, así como para cumplir con las demandas éticas y legales que reconocen a los animales como seres capaces de experimentar dolor y estrés.

Sin embargo, el monitoreo del bienestar animal presenta desafíos significativos. Las evaluaciones actuales se basan principalmente en inspecciones visuales, las cuales siguen siendo limitadas en términos de tiempo, recursos y precisión.

Ante esta problemática, se han propuesto metodologías que implementan el uso de sensores y la inteligencia artificial para el monitoreo automatizado del bienestar animal. Estas metodologías tienen el potencial de proporcionar mediciones continuas y precisas de variables indicadoras de bienestar, superando las limitaciones de las inspecciones visuales. Sin embargo, en el contexto específico del monitoreo de vacas en pastoreo libre, aún existe una necesidad de determinar cuál de estas metodologías es la más efectiva y eficiente.

En este sentido, surge la siguiente interrogante de investigación: ¿Cuál de los modelos de detección de objetos, específicamente el modelo pre-entrenado YOLO y los modelos entrenados para la detección de vacas, presenta el mejor desempeño en el monitoreo de vacas en pastoreo libre para el desarrollo de un sistema de monitoreo de bienestar animal?

El planteamiento de este problema de investigación busca determinar qué modelo de detección de objetos es el más adecuado para el monitoreo de vacas en pastoreo libre, con el fin de establecer las bases para el desarrollo de sistemas automatizados que mejoren la precisión y la eficiencia en la evaluación del bienestar animal en la industria ganadera. Esta investigación contribuirá a la selección de un modelo robusto y eficiente que sirva como base para futuros trabajos relacionados con el monitoreo de bienestar animal, impactando positivamente en la calidad de vida de los animales y promoviendo prácticas más sostenibles en la industria ganadera.

III. OBJETIVOS

3.1 Objetivo General

El objetivo general de este trabajo de grado es desarrollar y comparar modelos de detección de objetos basados en YOLOv5 para el monitoreo de ganado en pastoreo libre, con el fin de determinar cuál de estos modelos presenta el mejor desempeño en términos de precisión y eficiencia. El resultado de esta investigación servirá como base para futuros trabajos relacionados con el monitoreo automatizado del bienestar animal en la industria ganadera.

3.2 Objetivos Específicos

- Aplicar técnicas de clustering para seleccionar un dataset representativo y variado que permita etiquetar las imágenes utilizadas para el entrenamiento de los modelos.
- Aplicar técnicas de aumento de datos para enriquecer el dataset de entrenamiento, mejorando la variabilidad y generalización del modelo de detección de ganado.
- Aplicar transferencia de conocimiento para entrenar modelos basados en YOLOv5, adaptándolos al contexto específico del monitoreo de ganado en pastoreo libre.
- Comparar el desempeño de los modelos entrenados, evaluando su capacidad de detección y localización de ganado en diferentes escenarios y condiciones.
- Evaluar la robustez del modelo final seleccionado, realizando pruebas exhaustivas en diferentes condiciones y escenarios para verificar su adaptabilidad y confiabilidad.
- Disponibilizar el modelo mediante una API REST y un servidor web, facilitando su integración en sistemas y aplicaciones de monitoreo de bienestar animal.
- Destacar el modelo de detección de objetos con el mejor desempeño y su aplicabilidad en futuros sistemas de monitoreo de bienestar animal.

Al alcanzar estos objetivos, se espera obtener un modelo de detección de objetos preciso y eficiente que pueda ser utilizado como base para el desarrollo de sistemas automatizados de monitoreo de bienestar animal en la industria ganadera. Además, se brindarán recomendaciones para futuras investigaciones en este campo, contribuyendo así al avance y mejora de las prácticas ganaderas orientadas hacia el cuidado y bienestar de los animales.

IV. MARCO TEÓRICO

4.1 Métricas de evaluación

La matriz de confusión es una herramienta fundamental para evaluar el rendimiento de un modelo de clasificación, bien sea binario o multiclase. Esta herramienta podría definirse como un resumen de la cantidad de veces que un modelo clasifica correcta o incorrectamente las muestras de cada clase. Estos resultados son expresados en una tabla de 2x2 (o NxN en el caso de clasificación con más de dos clases) donde las columnas representan las clases reales y las filas representan las clases predichas por el modelo (o viceversa). Es así que, la diagonal principal de la matriz representa los valores verdaderos positivos (TP) y verdaderos negativos (TN), mientras que los valores fuera de la diagonal principal representan los falsos positivos (FP) y falsos negativos (FN).

- Verdaderos positivos (TP): detección correcta realizada por el modelo.
- Falsos positivos (FP): detección incorrecta realizada por el detector.
- Falsos negativos (FN): Un objeto verdadero perdido (no detectado) por el detector de objetos.
- Verdaderos negativos (TN): esta es la región de fondo que el modelo no detecta correctamente. Esta métrica no se usa en la detección de objetos porque dichas regiones no se anotan explícitamente al preparar las anotaciones.

Una vez se tienen estos resultados en forma de matriz, es más sencillo ver no sólo dónde ha acertado sino dónde ha fallado el modelo en sus predicciones. Es por esto que también se puede decir que la matriz de confusión representa la incapacidad del modelo para clasificar correctamente y, por lo tanto, la "confusión" en el modelo [9].

Otras métricas que se pueden derivar de la matriz de confusión incluyen:

Precision: es la capacidad de un clasificador para identificar solo objetos relevantes. Está dada por:

$$Precision = \frac{TP}{TP+FP}$$

Recall: es una métrica que mide la capacidad de un clasificador para encontrar todos los casos relevantes. Está dada por:

$$Recall = \frac{TP}{TP+FN}$$

F1-Score: es la media armónica de *Precision* y *Recall*. Está dada por:

$$F1Score = 2(\frac{Precision \times Recall}{Precision + Recall})$$

Otra herramienta importante para la evaluación de la precisión del modelo es la **Curva de Confianza F1**. La curva F1 representa la relación entre la precisión y la exhaustividad (recall) del modelo, y se utiliza para determinar el mejor umbral de clasificación que maximice la puntuación F1.

La precision mide la proporción de objetos detectados que son verdaderamente de la clase de interés, mientras que el recall mide la proporción de objetos verdaderamente de la clase de interés que son detectados por el modelo. En otras palabras, la precision se enfoca en cuántos de los objetos detectados por el modelo son relevantes, mientras que el recall se enfoca en cuántos de los objetos relevantes son detectados por el modelo.

La curva F1 se genera al variar el umbral de clasificación del modelo y calculando la precision y recall para cada umbral. El punto de la curva F1 que maximiza la puntuación F1 (que es la media armónica de precision y recall) se utiliza para determinar el umbral de clasificación óptimo.

Finalmente, se usa como herramienta de evaluación **la curva de aprendizaje**, la cual es una representación visual que muestra cómo el rendimiento del modelo varía en función del tamaño del conjunto de datos de entrenamiento. En una curva de aprendizaje típica, el eje x representa el tamaño del conjunto de datos de entrenamiento, mientras que el eje y representa el rendimiento del modelo, generalmente medido en términos de la precisión o el error.

La curva de aprendizaje puede ayudar a evaluar el desempeño del modelo de detección de objetos de varias maneras:

- Ayudar a identificar si el modelo está sobreajustando o subajustando. Si el modelo tiene un rendimiento muy bajo en el conjunto de entrenamiento y el conjunto de prueba, es probable que esté subajustando. Si el modelo tiene un rendimiento alto en el conjunto de entrenamiento pero un rendimiento bajo en el conjunto de prueba, es probable que esté sobreajustando.
- Ayudar a estimar el tamaño óptimo del conjunto de datos de entrenamiento. Si el modelo tiene un rendimiento bajo en el conjunto de entrenamiento y el conjunto de prueba, es posible que necesite más datos de entrenamiento. Si el modelo tiene un rendimiento alto

en el conjunto de entrenamiento pero un rendimiento bajo en el conjunto de prueba, es posible que se necesite un conjunto de datos de entrenamiento más pequeño.

- Identificar las áreas en las que el modelo necesita más atención. Por ejemplo, si el modelo tiene un rendimiento bajo en el conjunto de prueba, es posible que sea necesario mejorar la calidad de los datos de entrenamiento o ajustar los parámetros del modelo.

V. DATOS

El conjunto de datos utilizado para el entrenamiento, validación y prueba del modelo proviene de dos fuentes: una propia y otra externa de libre acceso.

El *set* de imágenes propio consta de 1.038 fotografías de alta calidad digital (4000px de ancho * 3000px de alto) que fueron captadas con una GoPro HERO8 Black. Las fotografías captaron al **ganado bovino** en pastoreo y fueron tomadas entre las 03:00pm y 05:00pm a diferentes razas de ganado y en diferentes tomas (principalmente aéreas).

Por su parte, el *set* de imágenes externo fue tomado de Open Images, un conjunto de datos público que tiene aproximadamente 9 millones de imágenes de buena calidad digital (1024px en su lado más largo), todas ellas autorizadas por Google Inc. bajo la licencia CC BY 4.0 y Creative Commons Attribution bajo la licencia CC BY 2.0.

En este *set* de imágenes se encuentran más de 11.000 imágenes de **ganado bovino** con características particulares que permiten hacer un mejor entrenamiento del modelo y, por ende, obtener mejores resultados:

- **En diferentes actividades:** caminando, pastoreo, descansando, durmiendo, entre otros.
- **En diferentes entornos:** en corral, en prado, en bosques, en cultivos de trigo, en pavimento, entre otros.
- **Con otros elementos:** personas, otros animales, autos, casas, entre otros.

VI. METODOLOGÍA

6.1 Preprocesamiento de los datos

6.1.1 Selección del Dataset Inicial y Train/Test Split

Para el entrenamiento y validación del modelo, se cuenta inicialmente con un conjunto de 1.038 imágenes provenientes de fotogramas de un vídeo. Sin embargo, se observa que estas imágenes son altamente similares entre sí debido a su origen común.

Con el fin de obtener un conjunto más representativo y diverso, se seleccionan las 250 imágenes más diferentes. Para lograr esto, se tiene en cuenta la diferencia horaria entre las fotografías (seleccionando aquellas imágenes con intervalos de tiempo más amplios entre sí) para lograr así capturar variaciones significativas en las condiciones de luz y sombra, como también en la posición del ganado y la cantidad de elementos presentes en cada imagen. Este proceso asegura que las imágenes seleccionadas representan la mayor cantidad de escenarios posibles, lo cual es fundamental para entrenar un modelo robusto y generalizable.

Una vez seleccionadas las 250 imágenes más diversas, se procede a su etiquetado utilizando *Label Studio* [10], donde se generan *labels* en formato *pascal_voc* y *yolo*. Durante este proceso, se decide no etiquetar aquellos elementos en las que el ganado está de espalda, cuando se superpone con otro elemento, cuando están demasiado lejos de la toma o cuando está parcialmente fuera de la imagen, ya que se considera que esto puede introducir distorsiones o ambigüedades en el modelo final.

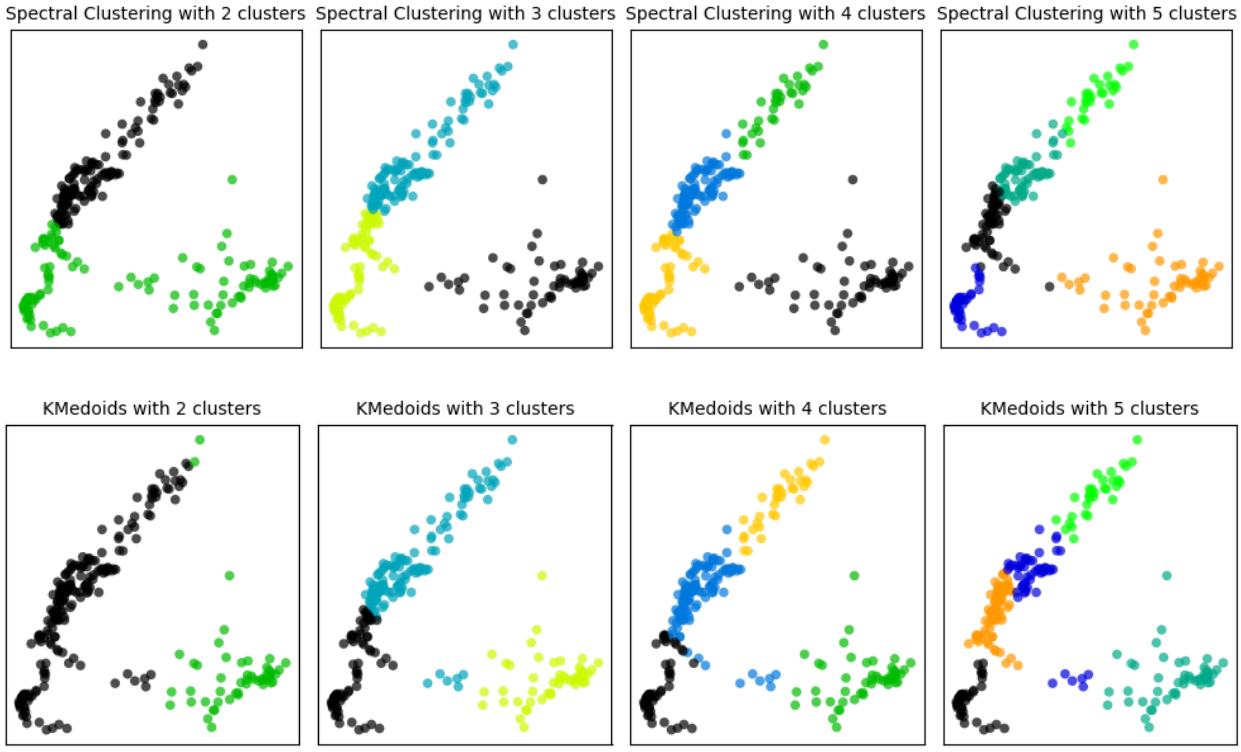


Figura 2. Ejemplo de algunos elementos que no se etiquetaron.

En la figura 2 se puede observar cómo se aplican los criterios de etiquetación mencionados anteriormente, los cuales se consideran importantes para el contexto y los objetivos del proyecto.

Después de tener el dataset inicial de 250 imágenes, se usan diversas técnicas (que se detallan a continuación) para elegir las 100 imágenes más diferentes para el conjunto de validación y las 150 restantes para el entrenamiento del modelo.

1. Se emplea la librería *pillow* para obtener características de las imágenes, como el promedio de color. Esta medida proporciona información valiosa sobre la distribución y la tonalidad general de los píxeles en cada imagen..
2. A continuación, mediante el uso de la biblioteca *scikit-learn*, se aplica una técnica de reducción de dimensionalidad conocida como Análisis de Componentes Principales (*Principal Component Analysis, PCA*). Esta técnica permite transformar las características de las imágenes en un espacio de menor dimensión, manteniendo la mayor cantidad de información relevante posible.
3. Posteriormente, se evalúan tres modelos de *clustering* utilizando *scikit-learn*: *KMedoids, KMeans y SpectralClustering*. Estos modelos se utilizan con el propósito de agrupar las imágenes en conjuntos que muestran características similares. Después de comparar los resultados, se evidencia que el rendimiento de los tres modelos es muy similar, por lo cual se decide escoger el algoritmo *KMeans*.



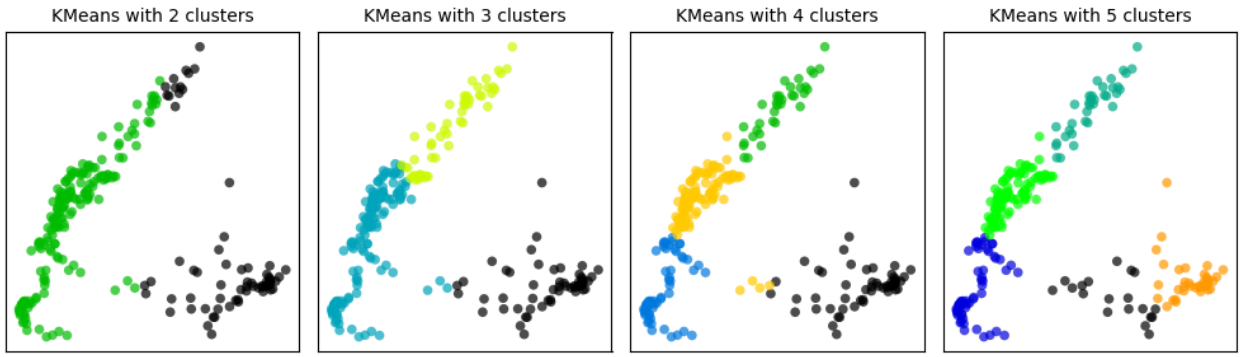


Figura 3. Comparación de los clusters formados después de aplicar los algoritmos Spectral Clustering, KMedoids y Kmeans.

4. Para determinar la cantidad óptima de *clusters* en *KMeans*, se aplica la técnica de análisis de silueta (*silhouette analysis*) proporcionada por *scikit-learn*. Esta técnica calcula una medida de cuán bien se agrupa cada imagen en su respectivo *cluster*, y se busca el número de *clusters* que maximice este valor. En este caso, el análisis de silueta indica que el número óptimo de *clusters* es 4.

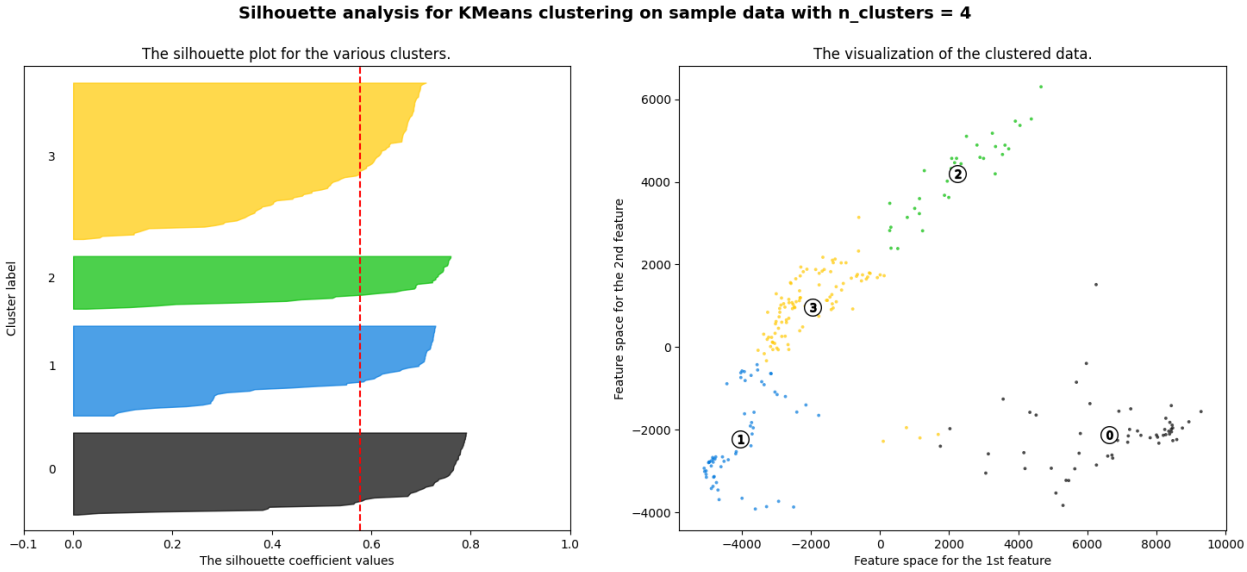


Figura 4. Análisis de Silueta que permite identificar que el número óptimo de clusters es 4.

5. Finalmente, utilizando la biblioteca *numpy*, para cada *cluster* se realiza un cálculo de la norma (distancia) de cada imagen con respecto a su centroide y se seleccionan las 25 imágenes más diferentes aplicando un enfoque basado en la distancia (equidistantes).

Este proceso de selección de imágenes permite que el modelo resultante esté expuesto a diferentes situaciones y escenarios, lo que probablemente se vea reflejado en una mejor robustez, capacidad de generalización y precisión en la detección del ganado bovino en condiciones reales.



Figura 5. muestra de imágenes que fueron incluídas en el Cluster 0.



Figura 6. muestra de imágenes que fueron incluídas en el Cluster 1.



Figura 7. muestra de imágenes que fueron incluídas en el Cluster 2.



Figura 8. muestra de imágenes que fueron incluídas en el Cluster 3.

En las muestras de imágenes de cada *cluster*, se puede observar que el algoritmo KMeans logra una separación efectiva al considerar las características visuales de cada imagen. Por ejemplo, en el *cluster 0* y en el *cluster 2* se encuentran imágenes con la presencia de sombras de árboles. Sin

embargo, en el caso del *cluster 0*, la sombra es más oscura, con mayor brillo y saturación, y solo se encuentran las ramas. Mientras tanto, en el caso del *cluster 2*, la sombra es más tenue debido a la hora de la foto y, además, se puede observar parte del tallo del árbol debido a un cambio en la posición del sol.

Por otro lado, el *cluster 1* y el *cluster 3* presentan similitudes, pero también se pueden apreciar algunas diferencias distintivas, siendo la posición de las vacas la más evidente; en el *cluster 1* hay una o más vacas cerca a la cámara, mientras que en el *cluster 3* todas están lejos de la toma.

6.1.2 Data Augmentation al Dataset de entrenamiento

Se emplea la técnica conocida como *Data Augmentation* para enriquecer y diversificar el conjunto de entrenamiento, el cual inicialmente consta de 150 imágenes.

Data Augmentation se refiere a la aplicación de transformaciones y modificaciones controladas a las imágenes existentes, con el objetivo de generar nuevas imágenes que capturen diferentes aspectos y variaciones de la realidad. Esta técnica es ampliamente utilizada en el campo del aprendizaje automático porque permite aumentar la diversidad y variabilidad del conjunto de entrenamiento, lo cual es crucial para enseñar al modelo a reconocer y adaptarse a diferentes condiciones y situaciones, evitando así problemas de sobreajuste y mejorando la capacidad de generalización del modelo a escenarios no vistos previamente. Según algunos estudios realizados, aplicando técnicas de *Data Augmentation* el rendimiento del modelo puede mejorar hasta en 1.5% [11].

A primera vista, *Data Augmentation* parece algo simple como aplicar transformaciones básicas a una imagen: espejo (*mirroed*), rotación (*rotated*), cambio en brillo o contraste (*brighter*), etc:



Figura 9 . Ejemplo de algunas transformaciones básicas.

Sin embargo, en el contexto de un modelo de Detección de Objetos es preciso contemplar otros aspectos para hacer más eficiente el aumento de imágenes:

- Aplicar la transformación a la imagen y generar nuevas etiquetas según las transformaciones/modificaciones aplicadas.
- Poder controlar la probabilidad y la magnitud de los valores para cada transformación.
- Generar un *pipeline* con diferentes transformaciones y probabilidades, porque generalmente no se aplica un único aumento por imagen.

Existen distintas librerías que pueden cumplir con estas necesidades, y entre ellas, *Albumentations* es ampliamente utilizada en el gremio. Esto se debe a que es una librería sencilla de implementar y que tiene una amplia gama de operaciones de transformación de imágenes. Además, está optimizada para brindar la máxima velocidad y rendimiento en su funcionamiento.

Algunos ejemplos comunes de los aumentos realizados mediante *Albumentations* incluyen la rotación de la imagen en diferentes ángulos para simular una variación en la posición de los elementos, el cambio de escala para representar diferentes distancias de captura, el ajuste del brillo, contraste y stauración para imitar condiciones de iluminación variables, la transformación del espacio de color para intensificar cualquier canal RGB, la aplicación de deformaciones elásticas para modelar cambios en la forma y la perspectiva de los objetos en la imagen, los filtros kernel para agudizar o desenfocar una imagen, el recorte aleatorio eliminar total parcialmente parte de los elementos, entre muchos otros. Todo esto con el fin de obtener un conjunto de datos enriquecido y más representativo de las diversas condiciones en las que puede encontrarse el ganado bovino.

Se define entonces un *pipeline* con 100% de probabilidad de aplicar hasta 18 transformaciones diferentes por imagen y se aplica al *dataset* de entrenamiento, pasando así de 150 a 1350 imágenes. Esta es una muestra de las posibles transformaciones aplicadas (la primera imagen es la original):



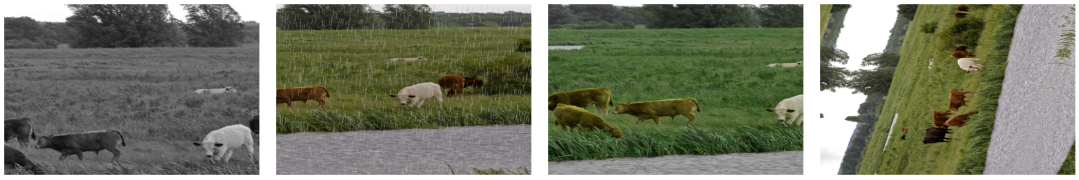


Figura 10. muestra de las posibles transformaciones aplicadas al conjunto de datos de entrenamiento.

6.2 Análisis de robustez de los modelos

Considerando que el conjunto de datos de validación usado para medir el desempeño de los modelos proviene de un solo escenario que, además, sólo contiene fotos tomadas entre las 3:00 y 5:00 pm, es necesario plantear una estrategia para determinar qué tan robusto es cada uno de los modelos, es decir, que tanto se ve afectado su desempeño ante el cambio de condiciones en la imagen que se provee como dato de entrada para la identificación de objetos.

La estrategia que se utiliza en este trabajo para medir la robustez, consiste en la modificación del conjunto de datos de validación aplicando diferentes transformaciones disponibles en la librería *Albumentations* para simular 9 escenarios de cambio de condiciones y posteriormente, evaluar el desempeño de los modelos evaluando dichos escenarios.

Cabe mencionar, que no todas las transformaciones utilizadas en este análisis fueron utilizadas en el proceso de aumento de datos aplicado a los conjuntos de datos de entrenamiento. Esto permite comparar el desempeño del modelo ante escenarios que se abordan en el entrenamiento respecto a escenarios que no son explícitamente incluidos en la distribución de dichos datos.

Específicamente, se aplican transformaciones que simulen los siguientes escenarios:

- *Downscale*: Busca simular condiciones en las cuales las imágenes de entrada tengan una menor resolución (El 20% de la resolución de las imágenes originales)
- *Fog:* Simula condiciones de niebla
- *Snow:* Simula presencia de nieve en las imágenes
- *Night:* Simula imágenes capturadas en la noche
- *Perspective:* Simula cambios en la perspectiva de las imágenes originales
- *Rain:* Simula condiciones de lluvia
- *Bright:* Simula un aumento en el brillo de la imagen

- *HueSaturation:* Simula cambios en los colores y la saturación de la imagen
- *Shadow:* Simula sombras aleatorias sobre las imágenes originales

En la figura 11 se observa un ejemplo de las imágenes transformadas para simular las condiciones de los escenarios *Downscale, Fog, Snow, Night* y *Perspective.* Estos escenarios no fueron utilizados en el proceso de aumento de datos realizado al dataset de entrenamiento. Un ejemplo del resto de los escenarios (*Rain, Bright, HueSaturation, Shadow*) se observa en la figura 12. Estos escenarios se aplicaron al dataset aumentado de entrenamiento.

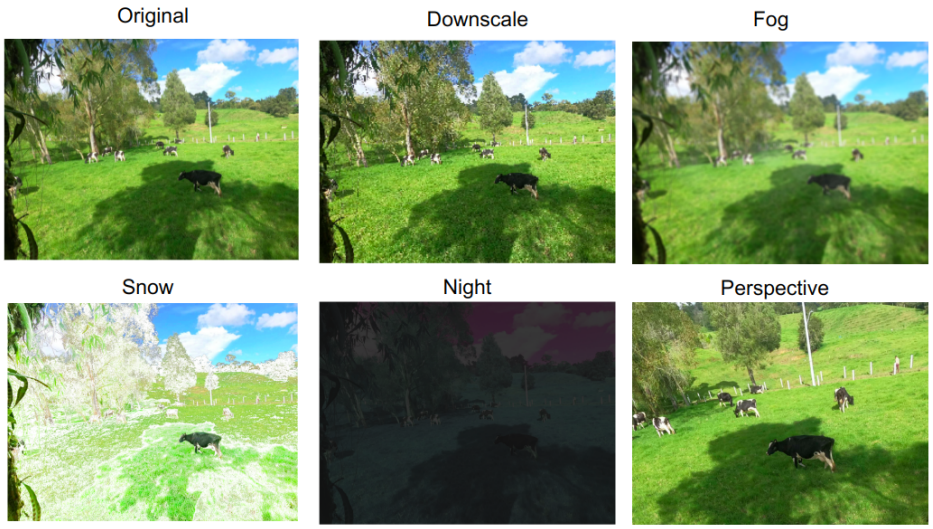


Figura 11. Ejemplo de las transformaciones aplicadas a una imagen en los escenarios Downscale, Fog, Snow, Night y *Perspective.*



Figura 12. Ejemplo de las transformaciones aplicadas a una imagen en los escenarios Rain, Bright, HueSaturation y *Shadow (Escenarios incluidos en el entrenamiento).*

6.3 Iteraciones para la selección del modelo

Como se ha mencionado anteriormente, en este trabajo se aborda un problema de detección de objetos. La detección de objetos es una importante tarea dentro de la visión artificial que busca identificar y localizar objetos específicos dentro de imágenes o videos. En el desarrollo de este trabajo se usa el detector YOLOv5 [7], el cual es uno de los principales algoritmos de detección de objetos de una sola etapa, que para el día de hoy, logra los mejores resultados superando otros algoritmos de detección de objetos por un gran margen.

Considerando que YOLO ofrece la posibilidad de utilizar modelos pre-entrenados, los cuales fueron entrenados usando el dataset COCO (Common Objects in Context) [8], en el desarrollo de este trabajo se define el siguiente proceso iterativo cuya representación gráfica puede observarse en la figura 13:

1. Se lleva a cabo la evaluación del desempeño del modelo pre-entrenado yolov5 sobre el conjunto de validación propio. Este es el modelo base y su desempeño corresponde al desempeño de referencia respecto al cual se comparan las siguientes iteraciones del modelo.
2. Se realiza el entrenamiento del modelo yolov5 con el dataset Open Images en la categoría *Cattle*. En este caso, al dataset Open Images se le aplican técnicas de aumento de datos para incrementar la diversidad y cantidad de los datos disponibles. Posteriormente, se comparan los resultados del modelo entrenado usando el dataset original, respecto al entrenado usando el dataset aumentado. Finalmente, ambos se comparan con el desempeño de referencia.
3. Se repite el numeral 2 usando el conjunto propio de imágenes.
4. Se construye un nuevo modelo que toma como base el modelo resultante del numeral 2 y se entrena nuevamente usando el conjunto de datos propios. Este proceso se realiza con los conjuntos de datos originales y aumentados. Nuevamente el desempeño de los modelos obtenidos en esta iteración se comparan con el desempeño de referencia.
5. Después de comparar el desempeño de todos los modelos disponibles, incluyendo yolov5 y los modelos construidos en las iteraciones, se elige el mejor. Las métricas que se usan para dicha comparación son la matriz de confusión, sus métricas *Recall*, *Precision* y *F1-Score* y la curva *F1-Confidence,* las cuales fueron descritas en el marco teórico.

6. Finalmente, se realiza el análisis para determinar la robustez del modelo ante cambios en las condiciones de los datos de entrada. Si el modelo logra detectar correctamente la categoría *Cattle* dentro las imágenes el 75% de las veces y en el análisis de robustez su desempeño no se ve afectado en más de 5 puntos porcentuales, se define que el modelo es apto para ser desplegado en producción, de lo contrario, deben repetirse las iteraciones cambiando las estrategias de construcción de los conjuntos de datos de entrenamiento.

Cabe mencionar que, en todos los casos las validaciones se realizan utilizando el conjunto de datos propios. Específicamente, las 100 imágenes que son seleccionadas con el ejercicio de clustering descrito en el apartado anterior.

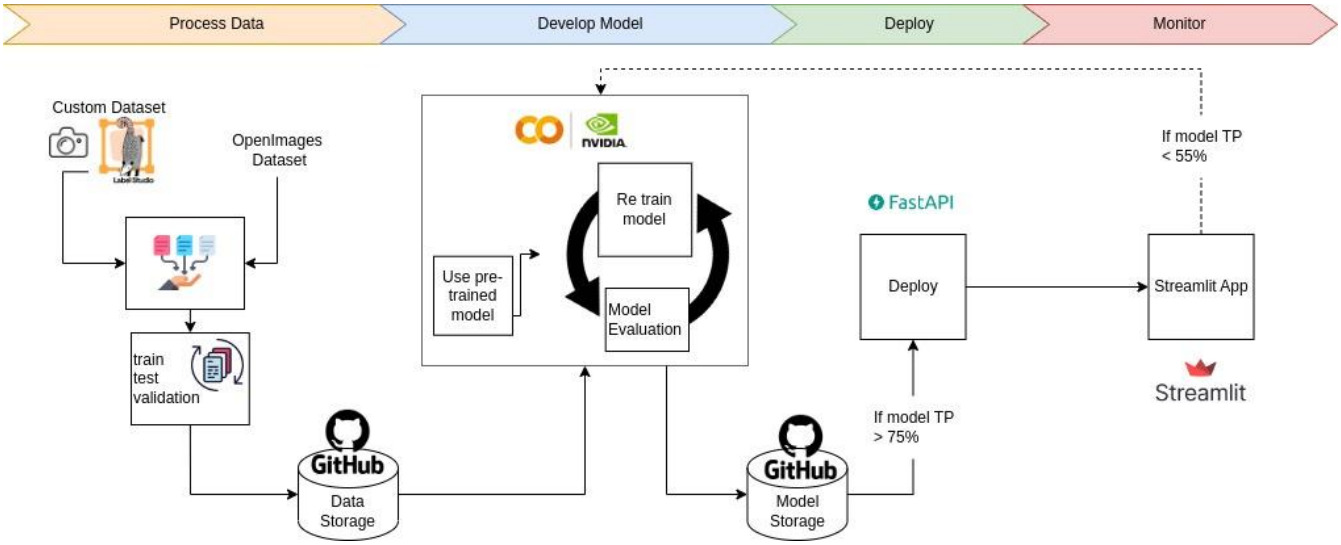


Figura 13. Iteraciones del modelo y despliegue en producción.

VII. RESULTADOS

7.1 Comparación del desempeño de los modelos

En total se realiza la comparación de siete modelos:

1. *yolo*: modelo pre-entrenado
2. *open_images*: modelo re-entrenado con el dataset de 150 imágenes descargadas de open_images pertenecientes a la categoría *Cattle*.
3. *open_images_aug*: modelo re-entrenado con el dataset de 1350 imágenes resultantes de la aplicación de diferentes pipelines de *data augmentation* a las imágenes descargadas de open_images pertenecientes a la categoría *Cattle*.
4. *custom_dataset*: modelo re-entrenado con el dataset propio de 150 imágenes.
5. *custom_dataset_aug*: modelo re-entrenado con el dataset de 1350 imágenes resultantes de la aplicación de diferentes pipelines de *data augmentation* al dataset propio.
6. *fine_tuned*: modelo que toma como base el modelo *open_images* y es entrenado nuevamente con el dataset propio de 150 imágenes
7. *fine_tuned_aug*: modelo que toma como base el modelo *open_images_aug* y es entrenado nuevamente con el dataset propio aumentado de 1350 imágenes.

En la figura 14 se observan las matrices de confusión de los siete modelos en cuestión y en la figura 15 se ilustra el desempeño de dichos modelos en términos de *Precision*, *Recall* y *F1-Score.* Los modelos *yolo*, *open_images* y *open_images_aug* detectan considerablemente más falsos positivos que los modelos entrenados usando el dataset propio, pero, a su vez, detectan más verdaderos positivos. Es por esto que son modelos con un *Recall* muy alto pero una *Precision* considerablemente menor. En la mayoría de los casos estos se puede explicar por las consideraciones mencionadas en la metodología, donde se explica que el ganado de espaldas o recortado no fue etiquetado, sin embargo, los modelos *yolo*, *open_images* y *open_images_aug* tienen la capacidad de detectar dichos objetos, resultado en esta cantidad de falsos positivos.

En la figura 16 se observan las curvas *F1-Confidence*, las cuales permiten ver el comportamiento del F1-Score al variar los niveles de confianza en el modelo. Para la evaluación de todos los modelos se fijó un valor de confianza de 0.5, sin embargo, valores entre 0.7 y 0.8 podrían maximizar el F1-Score sobre todo para los modelos *yolo*, *open_images* y *open_images_aug.*

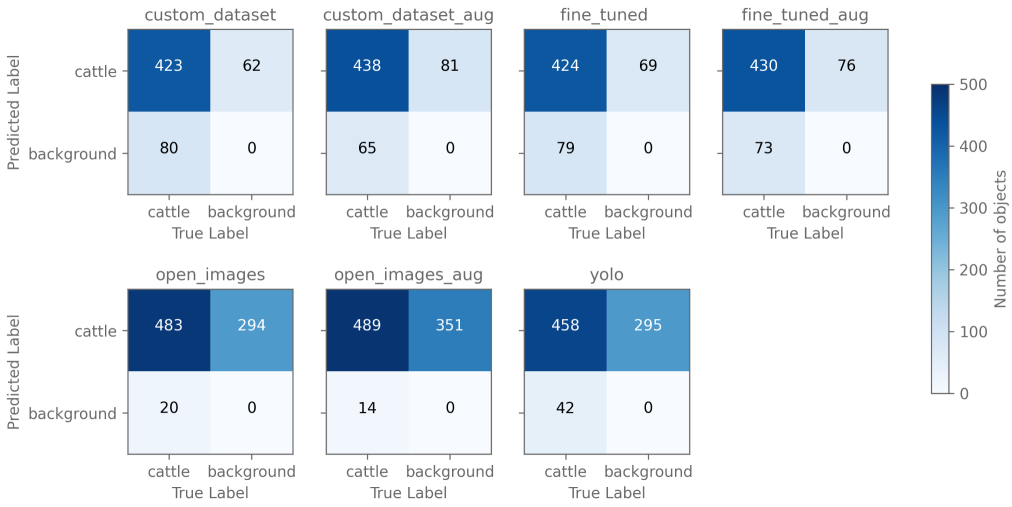


Figura 14. Matrices de confusión de los modelos evaluados.

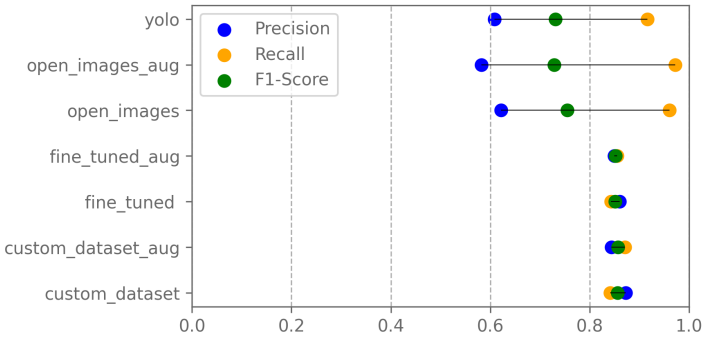


Figura 15. Métricas de desempeño de los modelos evaluados.

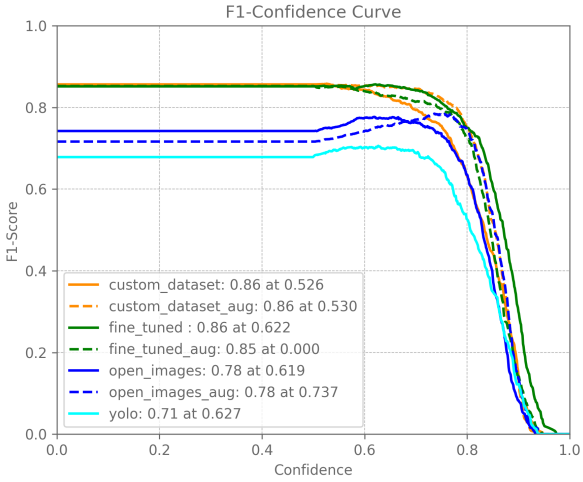


Figura 16. Curvas F1-Confidence de los modelos evaluados.

7.2 Resultados del análisis de robustez de los modelos

7.2.1 Análisis de robustez con escenarios que fueron usados en el entrenamiento

En la figura 17 se observa el desempeño de los modelos, en términos de su *Precision*, *Recall* y *F1-Score,* ante los escenarios de robustez *Rain, Bright, HueSaturation* y *Shadow.* Puede verse que los resultados obtenidos son similares a los que se presentaron con el dataset de validación original (figura 15), donde los modelos *open_images* y *open_images_aug* tienen valores de *Precision* relativamente más bajos. Esto puede ser debido a su capacidad de detectar más imágenes de las que fueron etiquetadas, lo que incrementa la cantidad de falsos positivos, como puede confirmarse en las figuras 18 a la 21 .

Los modelos *fine_tuned_aug* y *custom_dataset_aug* tienen valores más altos de *F1-Score* que los modelos *fine_tuned* y *custom_dataset* para los cuatro escenarios comparados aquí y especialmente, para el escenario *shadow* donde los últimos modelos presentan un aumento en la cantidad de falsos negativos. Esto demuestra la mejoría en el desempeño que puede tener un modelo si la distribución de los datos en su conjunto de entrenamiento es similar al conjunto de validación.

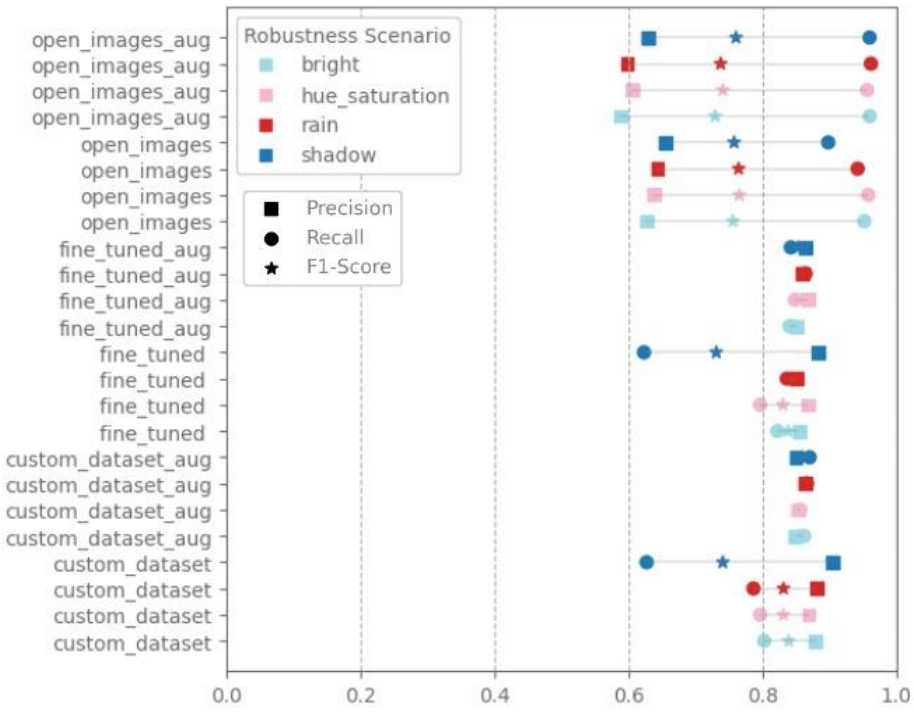


Figura 17. Métricas de desempeño de los modelos evaluados con los conjuntos de datos creados para medir la robustez de los mismos.

En la figura 22 se observan las curvas *F1-Confidence* para los 4 escenarios en cuestión. Puede verse que a pesar de que todos los modelos tienen resultados de *F1-Score* superiores a 0.7 para el nivel de confianza utilizado de 0.5, los modelos *open_images* y *open_images_aug* podrían maximizar su *F1-Score* usando un nivel de confianza cercano a 0.8.



Figura 18. Matrices de confusión de los modelos evaluados con el dataset que simula sombras aleatorias sobre la imagen



Figura 19. Matrices de confusión de los modelos evaluados con el dataset que simula lluvia.



Figura 20. Matrices de confusión de los modelos evaluados con el dataset que simula cambios en el color y saturación de la imagen.



Figura 21. Matrices de confusión de los modelos evaluados con el dataset que aumenta el brillo en las imágenes.

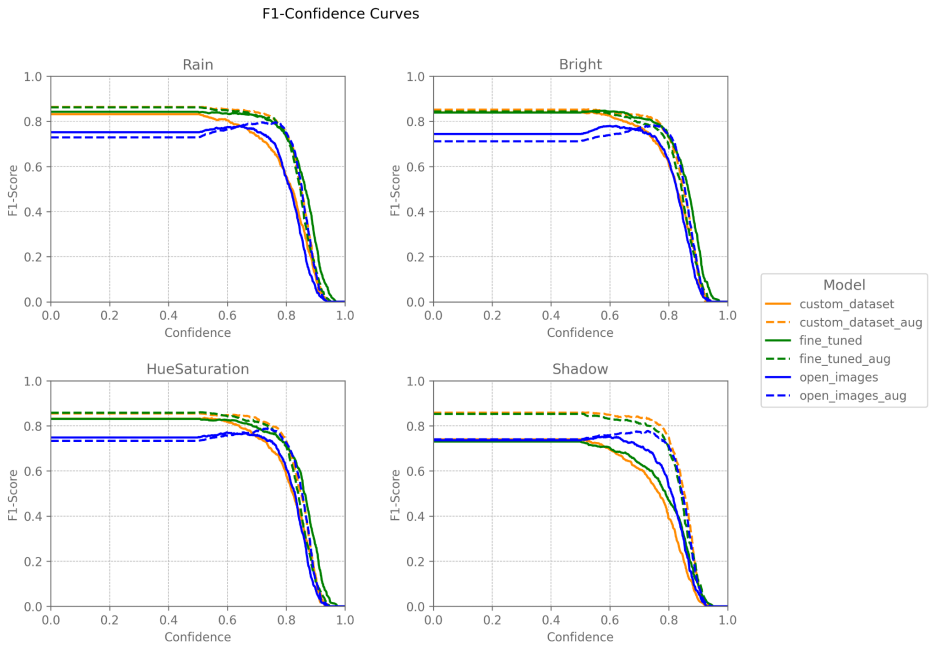


Figura 22. Curvas F1-Confidence de los modelos evaluados ante los escenarios Rain, Bright, HueSaturation y Shadow.

7.2.1 Análisis de robustez con escenarios que no fueron usados en el entrenamiento

En la figura 23 se observa el desempeño de los modelos, en términos de su *Precision*, *Recall* y *F1-Score,* ante los escenarios *Downscale, Fog, Snow, Night* y *Perspective.* Llama la atención como los modelos *open_images* y *open_images_aug,* a pesar de tener los valores de *F1-Score* más bajos con el dataset de validación original (figura 15) y con los escenarios de robustez usados en el entrenamiento (figura 17), son los que se ven menos afectados ante estos cambios de condiciones en las imágenes de entrada.

Por otro lado, los modelos *custom_dataset*, *custom_dataset_aug, fine_tuned y fine_tuned_aug,* a pesar de tener valores de *F1-Score* superiores a 0.8 en el dataset original y los escenarios *Rain, Bright, HueSaturation* y *Shadow.*, presentan una degradación en el desempeño ante algunos escenarios como niebla, nieve, noche y cambio en la perspectiva, en el que los valores de *F1-Score* llegan a valores incluso menores a 0.2.

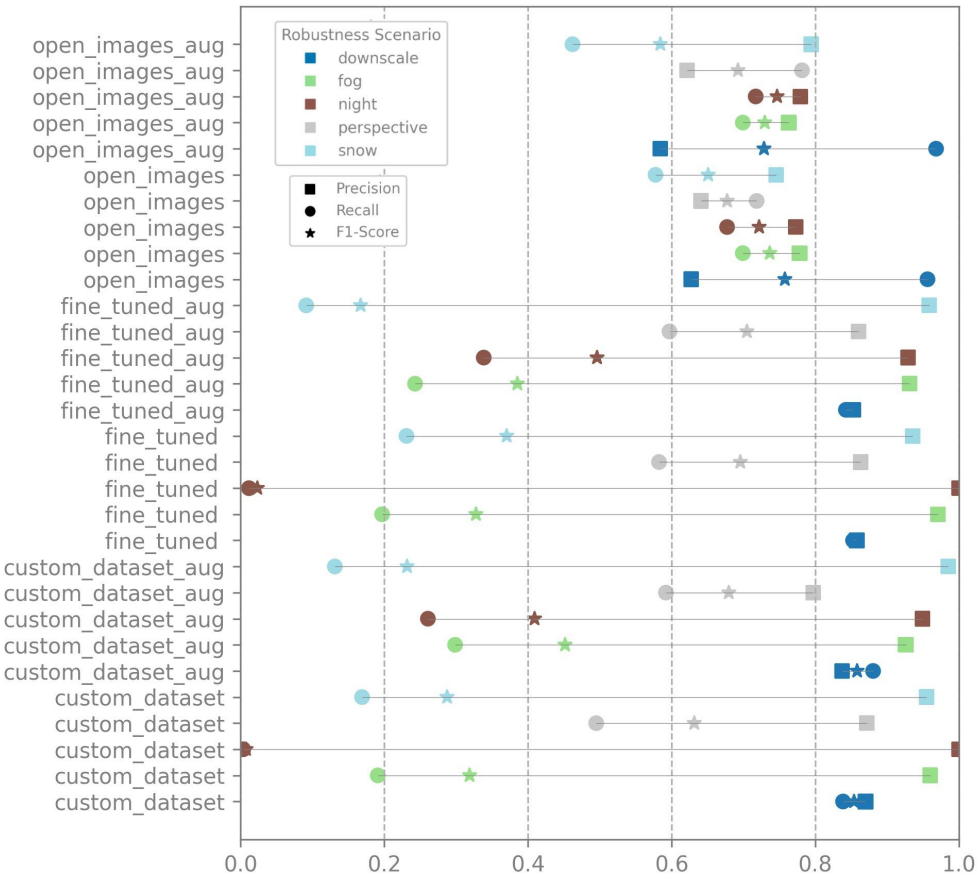


Figura 23. Métricas de desempeño de los modelos evaluados con los conjuntos de datos creados para medir la robustez ante los escenarios Downscale, Fog, Snow, Night y *Perspective.*

En las figuras 24 a la 28 se observan las matrices de confusión de los modelos ante los escenarios planteados para medir la robustez y en la figura 29 se observan las curvas F1-Confidence. Puede verse nuevamente como los escenarios niebla, nieve y noche, los modelos que maximiman la cantidad de verdaderos positivos son *open_images* y *open_images_aug,* y, además, son los modelos que permiten maximizar los valores de F1-Score para ser cercanos a 0.6 para un nivel de confianza de 0.5.



Figura 24. Matrices de confusión de los modelos evaluados con el dataset que simula condiciones de baja resolución.

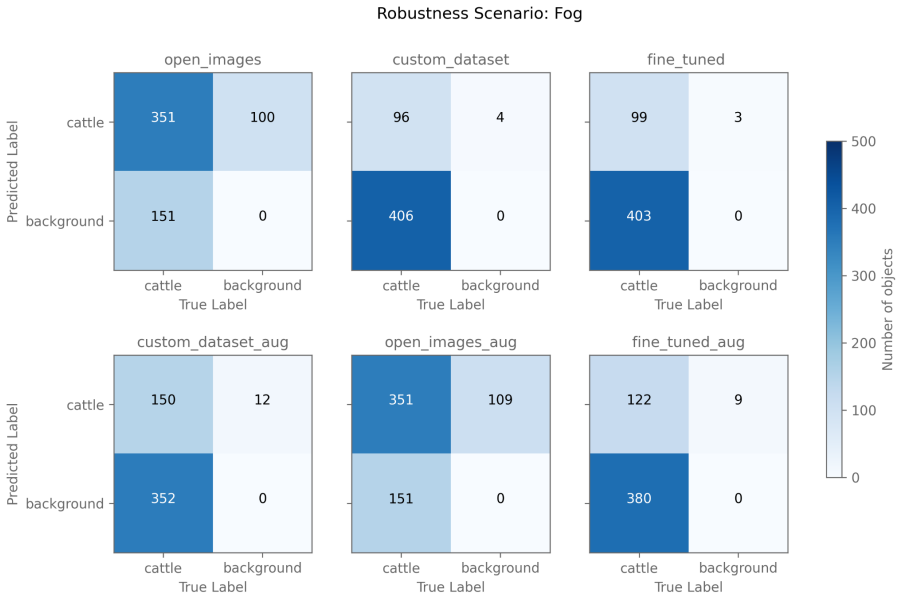


Figura 25. Matrices de confusión de los modelos evaluados con el dataset que simula condiciones de niebla.

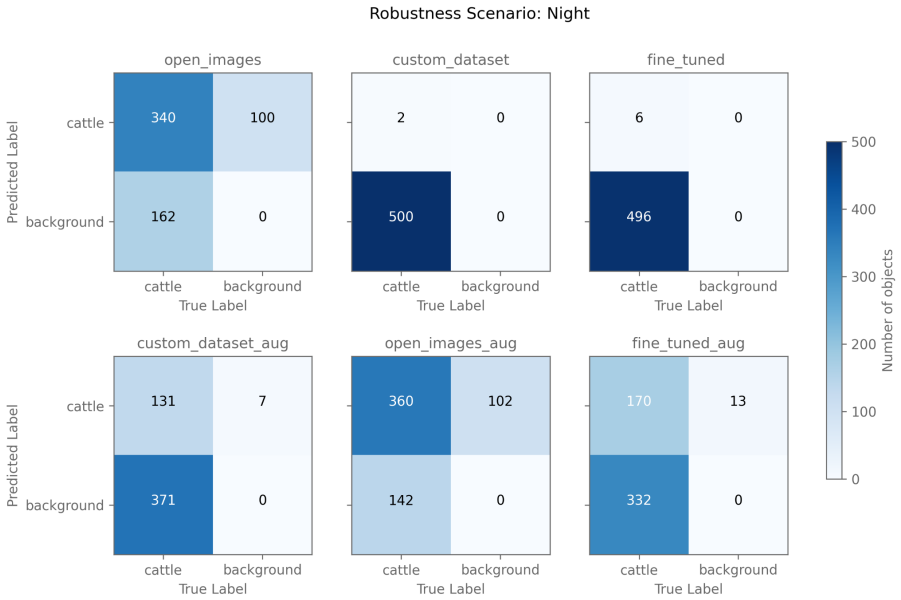


Figura 26. Matrices de confusión de los modelos evaluados con el dataset que simula imagenes capturadas en la noche.



Figura 27. Matrices de confusión de los modelos evaluados con el dataset que simula cambios en la perspectiva.

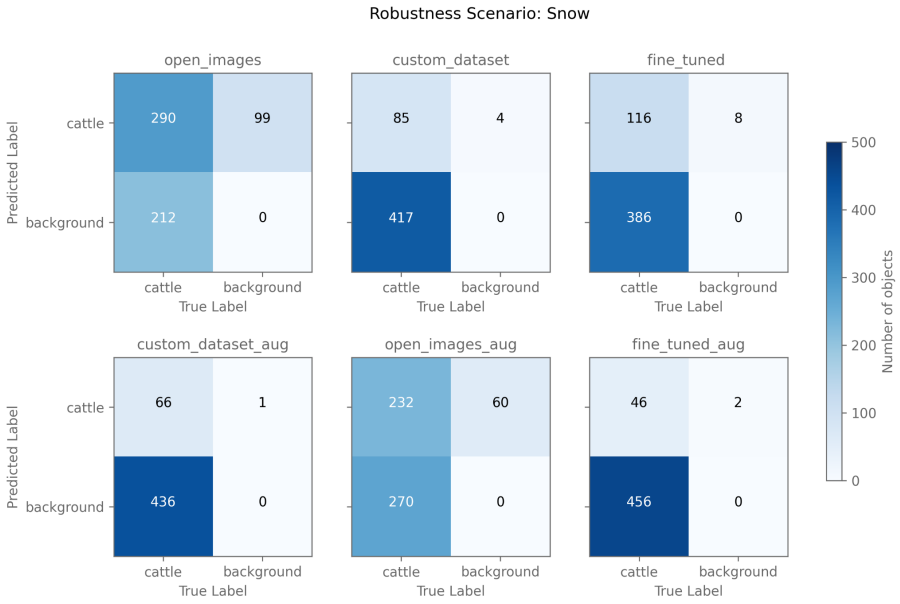


Figura 28. Matrices de confusión de los modelos evaluados con el dataset que simula condiciones de nieve.

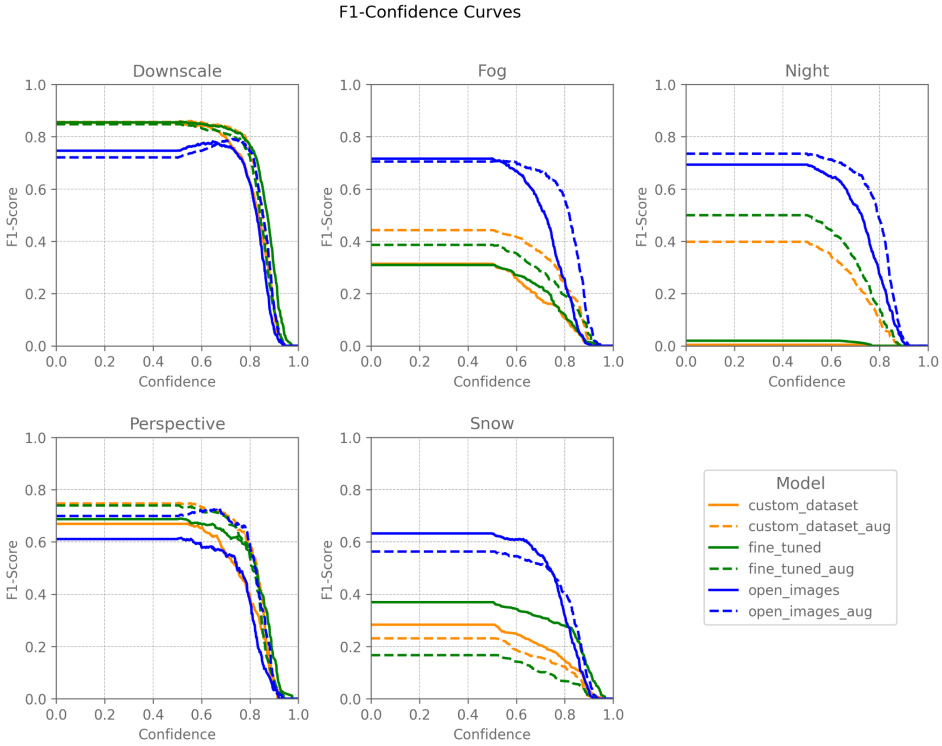


Figura 29. Curvas F1-Confidence de los modelos evaluados ante diferentes escenarios

VIII. BACKEND y FRONTEND

8.1 Backend

Se construye una API usando el framework **FastAPI** en Python.

Esta API tiene un método ***post*** que permite al usuario enviar una imagen y un texto al servidor.

La imagen ***(file)*** se recibe como un tipo *byte* y el texto (***weights***) como tipo *string*. Estos dos objetos se usan como argumentos dentro de la función para instanciar el modelo, hacer la predicción y finalmente retornar un resultado

8.2 Frontend

Se construye una WEB APP usando el framework de **Streamlit** en Python.

En esta *web app* se implementan ***Input widgets*** *(buttons widget, radio button widget and file uploader widget),* ***Data display elements*** *(Display object JSON string),* ***Text elements*** y ***Containers*** *(for show images).*

Esta *web app* consume la API usando la librería ***requests*** de Python enviando la imagen a analizar (*image*) y un argumento (*params*) que define el modelo a utilizar para la predicción**.**

La *web app* recibe respuesta de la API y muestra en pantalla: la imagen original, la imagen con los elementos detectados y un json con el resumen de los resultados:



Figura 30. Funcionamiento de la aplicación web

Todo el código relacionado se encuentra en: https://github.com/mauriciosierrav/cattle-detection.

IX. CONCLUSIONES Y RECOMENDACIONES

9.1 Conclusiones

- En este trabajo, se compararon siete modelos de detección de objetos para el monitoreo de ganado en pastoreo libre. Los modelos *yolo*, *open_images* y *open_images_aug* demostraron una mayor capacidad de detección de ganado, aunque con un mayor número de falsos positivos debido a su capacidad para detectar objetos en situaciones desafiantes.
- Los modelos entrenados con el dataset propio presentaron una detección más precisa, con menos falsos positivos con el conjunto de datos de validación original, sin embargo, podrían tener una menor capacidad de detección en escenarios adversos como niebla, nieve y noche.
- El ajuste de los niveles de confianza en los modelos *yolo*, *open_images* y *open_images_aug* permitió maximizar el F1-Score, indicando que establecer un umbral de confianza entre 0.7 y 0.8 podría mejorar su desempeño general.
- Los modelos *open_images* y *open_images_aug* demostraron ser más robustos ante cambios en las condiciones de las imágenes en escenarios no cubiertos en el proceso de entrenamiento, mientras que los modelos *custom_dataset*, *custom_dataset_aug*, *fine_tuned* y *fine_tuned_aug* mostraron una degradación significativa en estos escenarios adversos.

9.2 Recomendaciones

- Es importante considerar la robustez de los modelos frente a diferentes condiciones ambientales y escenarios adversos. Se recomienda evaluar el desempeño de los modelos en situaciones reales y recopilar *datasets* que abarquen una amplia gama de condiciones para mejorar la capacidad de detección en entornos desafiantes.
- Se sugiere explorar técnicas de aumento de datos adicionales para mejorar la robustez de los datasets de entrenamiento. Estas técnicas pueden incluir la simulación de condiciones adversas, para fortalecer la capacidad de detección de los modelos.
- Es importante seguir recopilando y expandiendo los datasets de entrenamiento para incluir una amplia variedad de condiciones ambientales y escenarios adversos. Esto permitirá mejorar la capacidad de detección de los modelos en entornos desafiantes y garantizar un mejor rendimiento en situaciones del mundo real.

Referencias

- [1] Tomás, M., & Gordillo, Á. (2022). Bienestar animal ¿Clave para la sostenibilidad en la producción porcina? *Bioseguridad en ganadería*. https://bioseguridad.net/rodenticida/bienestar-animal-clave-para-la-sostenibilidad-en-la-producion-porcina/

- [2] Neethirajan, S. (2020). The role of sensors, big data and machine learning in modern animal farming. *Sensing and Bio-Sensing Research*, *29*(100367). https://doi.org/10.1016/j.sbsr.2020.100367

- [3] Davidson, T., & ZIESCHE, S. (2021). The Overlaps Between Artificial Intelligence & Animal Welfare. *Philosophies*, *6(2)*(31). https://doi.org/10.3390/philosophies6020031

- [4] *Assessment Protocol for Cattle*. (2009). Welfare Quality Consortium. https://edepot.wur.nl/233467

- [5] de Jong, I.C., Hindle, V.A., Butterworth, A., & Engel, B. (2016). Simplifying the Welfare Quality® assessment protocol for broiler chicken welfare. *Animal*, *10*, 117-127. ScienceDirect. https://doi.org/10.1017/S1751731115001706

- [6] Tuyttens, F., de Graaf, S., Norlander, S., & de Boyer, A. (2021). Using Expert Elicitation to Abridge the Welfare Quality® Protocol for Monitoring the Most Adverse Dairy Cattle Welfare Impairments. *Frontiers in Veterinary Science*, *8*. https://doi.org/10.3389/fvets.2021.634470

- [7] Glenn, J., Nishimura, K., Mineeva, T., & Vilariño, R. (2020). yolov5. In *Code repository*. https://github. com/ultralytics/yolov5

- [8] Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, L. (2014). Microsoft COCO: Common Objects in Context. *European Conference on Computer Vision*, *8693*. SpingerLink. https://doi.org/10.1007/978-3-319-10602-1_48

- [9] Brownlee, J. (2016). What is a Confusion Matrix in Machine Learning. *Machine Learning Mastery*. https://machinelearningmastery.com/confusion-matrix-machine-learning/

- [10] Tkachenko, Maxim, Malyuk, Mikhail, Holmanyuk, Andrey, & Liubimov, Nikolai. (2020). Label Studio: Data labeling software. Retrieved from https://github.com/heartexlabs/label-studio

- [11] Garcia, M. (2020) Data Augmentation para la Clasificación Automática de la Calidad Vocal. *V Jornadas de Intercambio y difusión de los Resultados de Investigaciones de los Doctorados en Ingeniería.* https://doi.org/10.33414/ajea.5.748.2020